JOURNAL OF
EMERGING INVESTIGATORS

Article

Comparative study of machine learning models for water potability prediction

Helen Lee[1], **Lars Holdijk**[2]

[1] Saint Francis High School, Mountain View, California

[2] Lumiere Education, Oxford, United Kingdom

SUMMARY

Currently, water quality is an increasingly pressing issue globally because many people cannot access clean drinking water. In order to better predict the potability of water, scientists have used many machine learning models, such as artificial neural network (ANN) and support vector machine (SVM) models. However, many of these methods tend to be complex and take up a lot of computing resources, making them inefficient, so our research aimed to find a machine learning model that is not only effective at predicting the quality of water, but also simpler and more efficient. We hypothesized that neural networks would be the most effective at this task because of their ability to recognize patterns and underlying relationships within complex datasets. We experimented with four different machine learning models: logistic regressions, k-nearest neighbors, decision trees, and neural networks. Each algorithm was trained and validated using the same dataset. We found that logistic regression with L1 regularization had the highest precision score of 0.75000, and decision trees had the second highest precision score of 0.74359. When comparing the accuracy score, we found decision trees had a higher accuracy score than logistic regression. This could be due to the fact that L1 regularization estimates around the median of the data, while the "yes and no" structure of decision trees is very effective for binary classifications. As a result, we concluded that decision trees were the most effective at predicting water quality.

INTRODUCTION

According to the World Health Organization (WHO), in 2020, two billion people were unable to access safe drinking water, and this problem is expected to intensify due to the looming threats of climate change and water pollution (1). Nearly 1.5 million people died in 2019 due to diseases caused by drinking unsafe water (1). Many people have also been exposed to hazardous chemicals, such as lead, due to drinking chemically-contaminated water (1). To ensure that people can access safe drinking water, various methods have been developed and used to monitor and control water quality based on WHO published guidelines (2). These guidelines define the minimum requirements for safe drinking water in terms of biological, chemical, and physical water indicators (3). Biological water quality parameters include the presence and concentrations of bacteria, algae, viruses, and protozoa. Physical water quality parameters measure the physical aspects of water. These include pH, turbidity, temperature, color, solids, electrical conductivity, taste, and odor (3). Chemical parameters refer to the chemicals that can be measured from the water samples. There are more than 90 chemical contaminants listed in WHO published guidelines (3). Among them include chlorine, hardness (calcium and magnesium concentrations in water), sulfates, and trihalomethanes, which are some of the most frequently monitored parameters (4). When evaluating water quality, typically only a subset of parameters that have the most impact on human health is monitored (2). In this study, we focused on assessing nine water quality parameters: pH, hardness, total dissolved solids (TDS), chlorine, sulfates, conductivity, total organic carbon, trihalomethanes, and turbidity, which are the parameters that are most used to evaluate water quality (**Table 1**). These parameters can be used to indicate the presence of contaminants that can cause diseases and affect the taste and smell of the water.

Traditionally, water quality evaluation has been done manually for a given body of water, which is time-consuming and inefficient (5, 6). This is where machine learning can present a strategy for evaluating water quality in a more efficient manner. Machine learning has been increasingly used for water quality analysis in recent years because of its ability to handle vast, complex, nonlinear, relational data (2, 4). Researchers have developed and experimented with many models to demonstrate the ability of machine learning in monitoring and predicting water quality (2, 4-10). For example, a previous study used artificial neural networks (ANN) and support vector machines (SVM) to estimate water quality of rivers in Iran using ten parameters (5). Another study evaluated 1679 water samples in India, using seven water quality parameters and the following machine learning models: random forest (RF), multilayer perceptron (MLP), CATBoost, XGBoost, logistic regression, and decision trees (6). Among different algorithms, ANN and SVM have been widely used in evaluating water quality (2). These models are known for their short training time and high accuracy, especially when compared with other models such as RF and long short-term memory (LSTM) (2, 4). However, ANN and SVM are complex and often require significant computational resources to train which makes them, although accurate, less efficient (11-13).

In contrast, there are a number of less computationally intensive machine learning models. These include logistic regressions, k-nearest neighbors (KNN), decision trees, and neural networks. Logistic regressions estimate the relationship between two variables by using a linear model to classify data points into binary classes. They use the Maximum Likelihood Estimation (MLE) approach which pre-

Water Quality Parameter	Description of Each Water Parameter	WHO/EPA Recommended Values for Drinking Water
pH Level	pH is an important operational water quality parameter. It is an indicator of water acidity or alkalinity.	6.5–8.5 (WHO)
Hardness	Hardness is the level of calcium and magnesium in the water. The unit to measure hardness is milligrams per liter (mg/L).	< 500 mg/L (WHO)
Total dissolved solids (TDS)	Water can dissolve inorganic and organic minerals and salts. These minerals affect the taste and color of the water. TDS is an important parameter to measure water mineralization. The unit of measurement is mg/L.	< 600 mg/L, considered as good; > 1000 mg/L, not recommended (WHO)
Chlorine	Chlorine is used to disinfect water which is measured in mg/L.	< 5 mg/L (WHO)
Sulfate	Sulfates are substances found naturally in minerals. They can also be discharged by industrial wastes. The unit of measurement is mg/L.	< 500 mg/L (WHO)
Conductivity	Pure water does not conduct electricity well. Dissolved minerals increase the concentration of ions in the water which enhances the conductivity of the water. The conductivity of water is measured in microsiemens per centimeter (µS/cm).	< 400 µS/cm (WHO)
Total Organic Carbon (TOC)	TOC is used to measure the amount of organic compounds found in the water in the unit of mg/L.	< 2 mg/L (EPA)
Trihalomethanes (THMs)	THMs are chemicals found in water treated with chlorine. The levels of THMs in drinking water varies depending on the amount of organic matter present in the water. The unit of measurement is micrograms per liter (µg/L).	< 100 µg/L (WHO)
Turbidity	Turbidity is a measure of the light emitting properties of water and is used to indicate the quality of waste discharge with respect to colloidal matter. The unit of measurement for turbidity is Nephelometric Turbidity Units (NTU).	< 5 NTU (WHO)

Table 1: Water quality parameters used in this study. For each water parameter, the table lists its name, the description of each water quality parameter, and the suggested values for drinking water per the WHO or EPA (3, 16, 21).

sets the mean and variance of the data as parameters when defining the parameters for the model (13). Compared to other models like SVM, logistic regressions are far simpler, hence requiring less computational resources, making them easier to implement (4).

KNN is a model that makes predictions about data points based on the k-nearest data points around it by using the assumption that similar data points fall closely to each other. The k-value is defined as the number of data points around a specific point that the model evaluates. When selecting the optimal k-value, techniques like cross-validation are used to test different k-values. The k-value that maximizes the model's performance is selected. The distance between two data points is calculated by using certain distance metrics. The closer the two points are, the traits they represent tend to be more similar. Even after the k-value is set, the model can still be trained using new distance metrics. This algorithm is simple, effective, and able to capture the complex relationship between data. However, the computing resources needed for calculating the distance between data points greatly increase when the training data gets larger; hence, this model is more efficient with smaller datasets (13, 14).

Decision trees represent data in a tree structure. The features of the dataset are represented by nodes, a decision rule is shown by a branch, and each outcome is shown by a leaf node. Decision trees recursively partition the dataset into subsets based on splitting criterion until the data is categorized. They use the attributes selection measure (ASM) to select the best attributes, and then make them into nodes to build the "tree" and can identify and capture the underlying relationships between variables. When partitioning the dataset, information gain and Gini impurity are used as the methods to decide the optimal split from a root node, and subsequent splits from sub-nodes. Information gain evaluates the change in entropy which is the uncertainty in data as it is being split, while each node in the decision tree is split into smaller subsets. Gini impurity evaluates how often a randomly chosen element from the decision tree is inaccurate. Both methods can greatly increase the precision of the decision tree by measuring the amount of uncertainty associated with the splitting of the data and likelihood of inaccuracies among its predictions (12, 15). Decision trees are the simplest classification method and require less computing power than other machine learning models, such as RF, because they have lower time complexity (2, 12, 13, 15).

Neural networks create a layered structure with interconnected nodes, with each node having weight and bias. The activation function evaluates the output value of a node from its weighted input and bias, and then determines whether the node should be activated or not. It also introduces nonlinearity into the model, which allows neural networks to uncover the hidden nonlinear relationships between input and output. To improve the model prediction accuracy and overall performance, an optimizer is usually used to minimize the loss function during training. Loss functions measure how well a neural network performs; the lower the value, the better the model is performing. Neural networks have many hyperparameters that need to be tuned for the neural network to be effective. Among them, the number of layers is especially important because it can greatly affect the accuracy and precision of the model. Too few layers could cause the model to be underfitted, while too many layers could cause the model to be overfitted. Another important hyperparameter is the number of nodes in each layer, which has a large impact on the complexity of the data the neural network can analyze. Neural networks are flexible and good for large, complex datasets. However, they tend to require higher computing power compared to other machine learning algorithms (11, 13).

Based on our review of the literature, so far, there is a lack of research on using simpler and more efficient models for predicting water potability. In this study, we compared four relatively simple machine learning classification models – logistic regression, KNN, decision trees, and neural networks. Specifically, we evaluated the ability of these four algorithms to accurately classify water potability based on water quality metrics. Our goal was to identify the most accurate algorithm while balancing computational efficiency and model complexity by comparing the performance of the four models listed above. Our work aimed to solve the problem of predicting water quality by finding a method that is both accurate while also being as simple and efficient as possible using machine learning.

We hypothesized that neural networks would be the most effective at predicting water potability because of their ability to identify complex relationships between input and output variables. However, our study revealed that decision trees were the best at predicting water potability instead. This

https://doi.org/10.59720/24-143

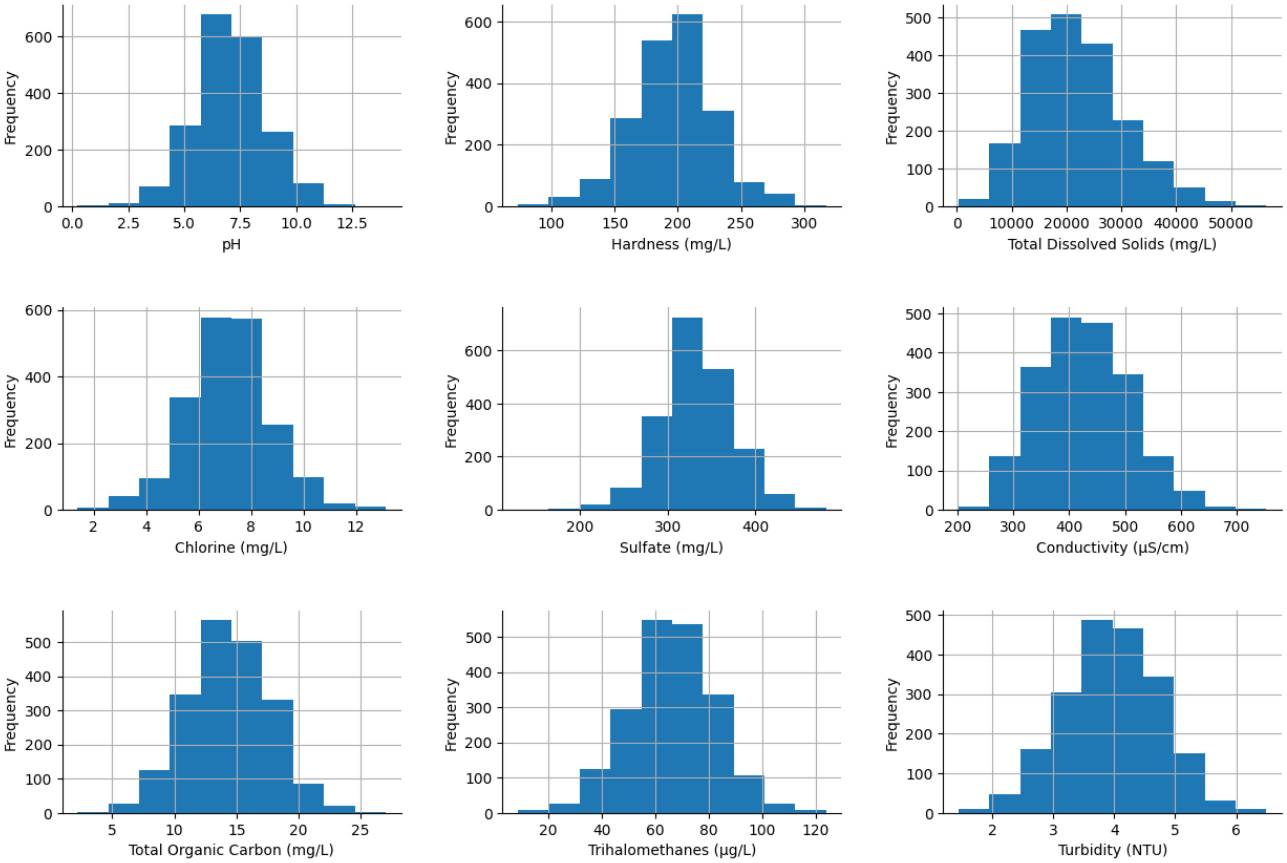


Figure 1: Data distribution for each water quality parameter. The x-axis represents the value of the feature, and the y-axis represents the number of data records for the given value. Each graph represents a single water quality metric being evaluated and each data point corresponds to a body of water in the dataset. The data for each feature is distributed symmetrically without outliers.

paves the way for a more efficient solution to the issue of detecting water potability.

RESULTS

We conducted our study on drinking water potability using a synthetic dataset published on kaggle.com (16). This dataset has nine features identified as critical metrics for safe drinking water according to the WHO drinking water standards (3), and has indicators of water quality from 3276 sources of water (16). Four machine learning models were trained and evaluated against the same dataset for all models. We randomly split the dataset into training and testing datasets according to a ratio of 70% training data to 30% testing data, which was 2293 samples in the training set and 983 samples in the testing set.

Solver	Penalty	5-Fold Cross-Validation Accuracy	10-Fold Cross-Validation Accuracy	Train Accuracy	Test Accuracy	Precision	Recall
liblinear	L1	0.59631	0.59774	0.60199	0.58444	0.75000	0.01186
saga	L1	0.59631	0.59774	0.60057	0.58444	0.75000	0.01186
lbfgs	L2	0.59418	0.59774	0.60128	0.58113	0.50000	0.00791
liblinear	L2	0.59418	0.59774	0.60199	0.58113	0.50000	0.00791
newton-cholesky	L2	0.59418	0.59774	0.60128	0.58113	0.50000	0.00791
sag	L2	0.59418	0.59774	0.60128	0.58113	0.50000	0.00791
saga	L2	0.59418	0.59774	0.60128	0.58113	0.50000	0.00791

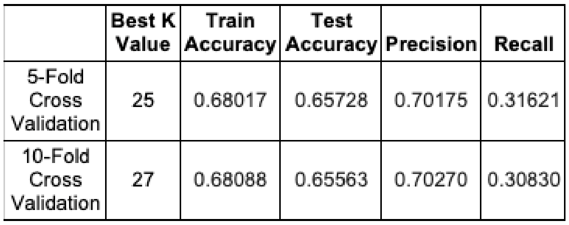
Table 2: Results of logistic regression. L1 (Lasso) regularizations with liblinear and saga optimization algorithms produce a 0.75000 precision score. L2 (Ridge) regulations with lbfgs, liblinear, newton-cholesky, sag and saga produce a 0.50000 precision score. The optimization algorithm (solver) which is used to optimize prediction output does not have an impact on the precision score calculation.

We created histograms from the dataset used in this study to analyze the distribution of the data for each feature. The histograms showed that each data had a symmetric unimodal distribution without any outliers (**Figure 1**).

Although the data we were evaluating was fairly symmetric, which decreased its likelihood of overfitting, to further prevent this occurrence from happening, we used k-fold cross-validation to enhance the model's ability to predict the water potability by partitioning the dataset into k number (5 and 10 in our experiments) of folds or subsets, then trained the model with k-1 folds and tested it with the remaining fold. Because cross-validation trained the model on many training sets, the model could predict the new data more accurately and precisely.

We evaluated the model performance with precision and recall. The precision, in this study, indicated that of all water sources that were classified as potable, what percentage of them were actually potable (true positives). The recall was the percentage of potable water sources that were predicted to be potable, as opposed to being classified as non-potable.

https://doi.org/10.59720/24-143



	Best K Value	Train Accuracy	Test Accuracy	Precision	Recall
5-Fold Cross Validation	25	0.68017	0.65728	0.70175	0.31621
10-Fold Cross Validation	27	0.68088	0.65563	0.70270	0.30830

Table 3: Results of k-nearest neighbors (KNN). KNN produces the similar results for 5-fold cross-validation and 10-fold cross-validation when the best k value is used to train the model for each cross-validation. A 0.70175 and 0.70270 precision score were produced respectively.

Logistic regression uses L1 (Lasso) and L2 (Ridge) regularization to mitigate overfitting. The difference between L1 and L2 is that L1 adds the absolute value of the coefficient as a penalty term to the loss function while L2 regularization adds the squared magnitude of the coefficient as a penalty term (13). Our findings show that L1 regularization with LIBLINEAR and SAGA optimization algorithms produced a 0.75000 precision score. L1 produced a 0.01186 recall score (**Table 2**). When L2 was used, even with the same optimization algorithms as L1, the model had a lower precision score of 0.50000 and a lower recall score of 0.00791.

KNN produced a 0.70175 precision score for 5-fold cross-validation and a 0.7027 precision score for 10-fold cross-validation, which means around 70% of the time, the model correctly predicts that the water is potable. The model also produced a 0.31621 recall score and a 0.3083 recall score for 5-fold cross-validation and 10-fold cross-validation respectively (**Table 3, Figure 2**).

When evaluating decision trees, we used information gain and Gini impurity as the methods to decide the optimal split from a root node and subsequent splits from sub-nodes. The results showed that decision trees produced a 0.74359 precision score and a 0.22925 recall score when using information gain, like entropy and log loss, as splitting criteria with 10-fold cross-validation. It was higher than training with 5-fold cross-validation. In comparison, the model produced a lower precision score when using Gini impurity as the splitting criterion for both 5-fold cross-validation and 10-fold cross-validation (**Table 4, Figure 3**).

We designed neural network models with the architecture containing between three and six layers. Each model consisted of the input layer, hidden layers, and the output layer (13). The input layer had nine neurons corresponding to the nine features defined in the dataset. The output layer had one node for the binary classification result. The number of neurons per hidden layer was the multiple of 16. We trained the dataset for 500 epochs with a 128 batch-size per epoch. The results showed that the models with more layers and more neurons on each layer produced higher precision scores than the models with fewer layers and fewer neurons on each layer **(Table 5)**. The highest precision score was produced by the 6-layer model which had the number of neurons in each layer as (9 (input layer) – 128 – 256 – 256 – 128 (hidden layers) – 1 (output layer)). This model had a precision score of 0.66667 and a recall score of 0.04743, which means that the model correctly predicts that the water is potable about 67% of the time (**Table 5**).

DISCUSSION

The parameters used in this study to measure water potability are based on the WHO and United States Environmental Protection Agency (EPA) published guidelines for safe drinking water quality. These parameters are pH, hardness, total dissolved solids (TDS), chlorine, sulfates, conductivity, total organic carbon, trihalomethanes, and turbidity (**Table 1**). The dataset consists of 3276 water sources (16). Compared with similar research done previously with a size range from several hundred to more than 20000 data points, the amount of data we used in our experiments is smaller (2).

We used accuracy, precision and recall to assess model performance. In machine learning, there is usually a trade-off between precision and recall. For a model to have a very high precision value, it needs to maximize the number of data points that are predicted to be positive and are actually positive. To ensure this, the model often predicts very few values to be positive to increase its precision score. However, this consequently also leads to a decrease in recall. As a result, it is very hard for recall and precision to increase simultaneously (11, 13).

Precision was extremely important to our study because we wanted every single water source the model determines to be potable to actually be potable. Otherwise, a false positive could cause people to become sick if they drink unpotable water that was deemed to be potable. As a result, the metric we evaluated the most to determine whether a model is effective or not was precision over recall. Recall is a measure of how well the model can detect potable water sources out of all of the actual potable sources. However, this measure was not as important because even if a water source turns out to

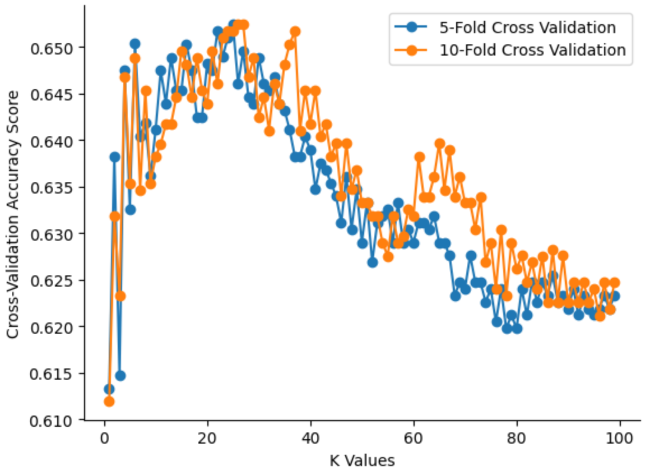


Figure 2: Correlation between the cross-validation accuracy score and the number of nearest neighbors used for K-Nearest Neighbor (KNN) model. The x-axis represents the number of neighbors used for k-neighbors queries, and the y-axis represents the cross-validation accuracy score. The line graph shows the trends of accuracy scores over k values for 5-fold cross-validation and 10-fold cross-validation. The Pearson r correlation coefficients are -0.77 for 5-fold cross-validation and -0.70 for 10-fold cross-validation. 5-fold cross-validation produced the highest accuracy score when the k value was 25. 10-fold cross-validation had the highest accuracy score when the k-value was 27. These best k-values were used to train the model. The results are listed in Table 3.

https://doi.org/10.59720/24-143

	criterion	max_depth of tree	Train Accuracy	Test Accuracy	Precision	Recall
5-Fold Cross Validation	entropy	8	0.77896	0.63245	0.58857	0.40711
5-Fold Cross Validation	gini	8	0.81805	0.63576	0.58824	0.43478
5-Fold Cross Validation	log_loss	9	0.80597	0.65232	0.63694	0.39526
10-Fold Cross Validation	entropy	3	0.66738	0.64404	0.74359	0.22925
10-Fold Cross Validation	gini	8	0.81805	0.64073	0.59574	0.44269
10-Fold Cross Validation	log_loss	3	0.66738	0.64404	0.74359	0.22925

Table 4: Results of decision tree. Decision tree produces a 0.74359 precision score when using information gain, such as entropy and log loss, as splitting criteria with 10-fold cross validation. When training with 5-fold cross-validation, it produces a 0.58857 precision score for entropy and a 0.63694 precision score for log loss. When using Gini impurity (gini) as the splitting criterion, the model produces a 0.59574 precision score for 10-fold cross-validation and a 0.58824 precision score for 5-fold cross-validation.

be unpotable, there are purification methods that can be used to make it drinkable, but if a person drinks from a water source deemed potable when it is, in actuality, not potable, it could have adverse consequences such as illness.

We hypothesized that neural networks would provide the most precise evaluation of water potability, because of their strength in identifying complexities between data values. However, we found that logistic regressions with L1 regularization produced the highest precision score of 0.75000, followed by decision trees with a precision score of 0.74359, contrary to our hypothesis. This could be because L1 regularization estimates around the median of the data when calculating the penalty to the loss function, which eliminates unimportant features. As a result, this also enhances the precision of logistic regression models, making them more precise compared to neural network models (13).

The decision trees had the second highest precision score, likely because the dataset used was small, fairly balanced and easier to interpret. As the amount of data increases, the decision tree becomes more complex which leads to more noise and hence more overfitting. As a result, decision trees are extremely effective with relatively smaller size datasets (12). The structure of a decision tree consists of the model asking "yes or no" questions to sort the dataset. It is known to work very effectively for binary classification which sorts the dataset into two categories, especially considering that the dataset was symmetric without outliers making it unlikely to be overfitted (12, 13). We also used cross-validation during the model training to find the maximum depth of the tree that produced the highest cross-validation accuracy score, and then used this optimal value to train and evaluate the model. This process helped prevent overfitting (17).

When comparing the accuracy score of the logistic regressions and decision trees, we found that decision trees had a higher accuracy score of 0.64404 compared to logistic

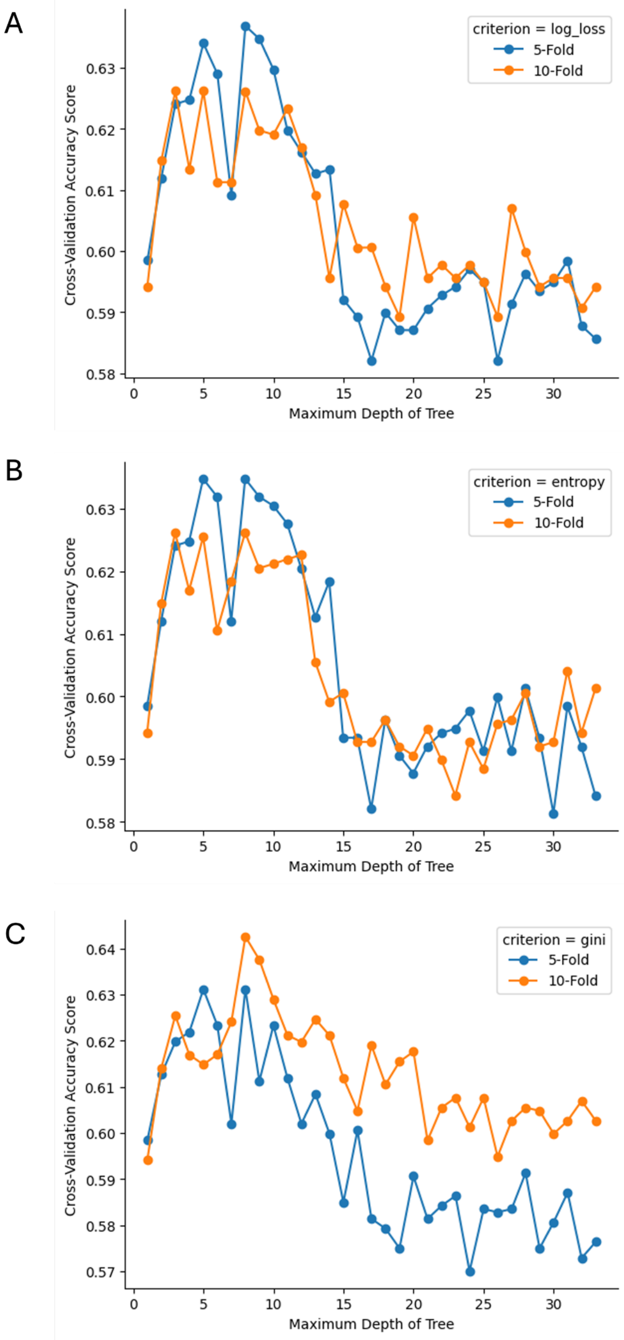


Figure 3: Correlation between decision tree cross-validation accuracy score and the maximum depth of the tree for each splitting criterion. The x-axis represents the maximum depth used for the decision tree, and the y-axis represents the cross-validation accuracy score. The line graph shows the trends of accuracy scores over the maximum depth of the tree for 5-fold cross-validation and 10-fold cross-validation for decision trees using log loss (A), entropy (B), or gini (C) as the splitting criterion. The Pearson r correlation coefficients for log loss, entropy, and gini are -0.75, -0.73, -0.82 for 5-fold cross-validation, and -0.71, -0.68, -0.58 for 10-fold cross-validation respectively. The maximum depth of the decision tree that produces the highest accuracy score is 9 for log loss, 8 for entropy, and 8 for gini with 5-fold cross-validation. For 10-fold cross-validation, the best maximum depth is 3 for log loss, 3 for entropy, and 8 for gini. The best values of the maximum depth are used to train the model. The results are listed in Table 4.

regressions which had an accuracy score of 0.58444. Hence, decision trees had a higher rate of correctly predicting water potability.

Overall, the precision scores were low for every model used in this study comparing to the results from previous studies using the ANN model (0.9206) and the SVM model (0.9197) (4). One possible reason for this trend is that the dataset used to train the models was relatively small compared to a larger sized dataset of over 10000 samples used based on previous studies done on this topic (2). In future studies, we could increase the ratio of training data to test data by splitting the dataset into the ratio such as 75 (training dataset) /25 (testing dataset), 80/20, or 85/15 with the hope that it would improve its precision score. However, we suspect that choosing less computationally-intensive models would result in a lower precision score compared with more computationally intensive models such as ANNs and SVMs. In the future, we could compare our current models' performance against the performance of more computationally intensive models. Another possible reason is that the experiments were conducted with mostly standard hyperparameters provided by the machine learning libraries. To improve the precision of each model, further research on this topic could involve more extensive hyperparameter tuning for all the models, as well as running all models against a larger dataset. Also, we could experiment with more advanced data preprocessing methods. In addition, future studies could include in-depth research on the models that worked the best in this experiment, like logistic regressions and decision trees; and research why these models were able to perform successfully when it came to predicting water potability compared with other models. Through conducting this research, we were able to conclude that decision trees are an effective method of predicting water potability through their higher precision and accuracy score. By evaluating efficient machine learning models that are less computationally-intensive, our findings bring us one step closer to solving the problem of inefficient and computationally-intensive methods in evaluating water potability.

MATERIALS AND METHODS

Dataset and Preprocessing

The dataset used in this project was published on kaggle.com on April 25, 2021. This dataset is updated annually and is currently on the third version. 3276 water sources were used in this dataset with each line of data representing 1 water body matrix (16). There were 3276 lines of data in this dataset in 10 data fields. The first nine fields contained the data for the nine water parameters: pH level, hardness, total dissolved solids (TDS), chloramines, sulfate, conductivity, total organic carbon (TOC), trihalomethanes, and turbidity (**Table 1**). The tenth data field was binary to indicate whether the water is safe for human consumption with 1 meaning water potable and 0 meaning not potable. This dataset was also synthetic, which means the data was randomly generated within a certain range of values (16).

The dataset contained null values in the fields of pH level, sulfate, and trihalomethanes, which could not be processed by the machine learning models. Instead of filling in the null values with the mean or median of that column's values, we removed the rows containing null values (n=1265) during the data preprocessing step. While this reduced the size of the dataset, we believed that having a larger dataset with random data backfilled would potentially create false positives regarding water potability. This was something we wanted to avoid because falsely marking unsafe drinking water as safe could potentially cause people to get sick.

Number of Layers in Model	Number of Nodes in Each Layer	Train Accuracy	Test Accuracy	Precision	Recall
3 Layers	9 - 16 - 1	0.58067	0.55795	0.45679	0.29249
4 Layers	9 - 16 - 16 - 1	0.58849	0.57781	0.48684	0.14625
5 Layers	9 - 16 - 16 - 16 - 1	0.61265	0.59106	0.57895	0.08696
6 Layers	9 - 16 - 32 - 32 - 16 - 1	0.61265	0.58775	0.54000	0.10672
6 Layers	9 - 32 - 64 - 64 - 32 - 1	0.60625	0.58609	0.61538	0.03162
6 Layers	9 - 64 - 128 - 128 - 64 - 1	0.61976	0.60265	0.60317	0.15020
6 Layers	9 - 128 - 256 - 256 - 128 - 1	0.61407	0.59106	0.66667	0.04743
6 Layers	9 - 256 - 512 - 512 - 256 - 1	0.61905	0.59106	0.61538	0.06324

Table 5: Results of neural networks. Overall, the neural network models with more layers and more neurons on each layer produce higher precision scores than the models with fewer layers and less neurons on each layer. The highest precision score, 0.66667, is produced by the 6-layer model (9 - 128 - 256 - 256 - 128 - 1).

Computing Resources

A Python 3 Google Compute Engine with a RAM of 12.57 GB and a disk with 107.72 GB was used in this experiment to run machine learning algorithms. When running neural networks related experiments, the T4 GPU was used. Other models were run using CPU. Python (v3.10.12) was used for this study (18). The machine learning libraries used were scikit-learn (v1.3.2) and PyTorch (v2.3.1) (19, 20).

Performance Metrics

The metrics used to evaluate the effectiveness of each machine learning model were accuracy, precision, and recall. Accuracy determines the number of data points evaluated by the model that are correct. Precision measures the proportion of true positive predictions among all the positive predictions made by the model, where precision = (true positives) / (true positive + false positives). Recall measures the proportion of true positive predictions among all actual positive cases in the dataset evaluated, where recall = (true positives) / (true positives + false negatives).

Cross Validation

In this experiment, cross-validation was used to further enhance the model's ability to predict the potability of water. We used k-fold cross-validation in this study, where we divided the data into k number of subsets or folds and used (k-1) of those folds to train the model and the remaining fold to evaluate the model. This process was repeated k times. After cross-validation, the results of each iteration were averaged to get a more precise approximation of the model's effectiveness. We used k-fold values of 5 and 10 (17).

Model Design and Training

To study the applicability and performance of machine learning in water potability classification, we tested four models: logistic regression, k-nearest neighbor, decision trees, and neural network. When each model was being run, the data was split into the training and testing datasets at a ratio of 70% training data to 30% testing data. We fit the model with the training dataset to learn the relationships between the variables and then validated the model with the testing dataset.

We used the LogisticRegression classifier from the library Scikit-learn to train the model, and experimented with two hyperparameters, solver and penalty. For the rest of the hyperparameters, we used the default values. The solver was the algorithm used to optimize prediction output. There were several options: "newton-cg", "lbfgs", "liblinear", "sag", and "saga". The penalty, also called regularization, was used to decrease the generalization error and control overfitting (19). There are two types of regularization, L1 (Lasso) regularization and L2 (Ridge) regularization supported by the library. To evaluate the impact of solver and penalty on the logistic regression performance, we ran the analyses with the different combinations of solver and penalty (**Table 2**).

We used KNeighborsClassifier from the Scikit-learn library with the default settings except for "n_neighbors". n_neighbors is the hyperparameter (k-value) to define the number of "neighbors" or close data points to look for in the dataset (19). To find the optimal k-value, we ran the cross-validation against the model and the training dataset over the possible k-values ranging from 1 to 100. After that, we identified the k-value which produced the highest cross-validation accuracy score and applied it to KNN model training and validation. As there were two commonly used k-fold values (5-fold and 10-fold) for cross-validation, we decided to test them both and compare the results (**Table 3, Figure 2**).

DecisionTreeClassifier from the Scikit-learn library was used for model training. We experimented with two hyperparameters, the maximum depth of the tree and the splitting criterion. The maximum depth of the tree was the number of decisions that the tree was allowed to make before coming to a classification. We ran cross-validations with 5-fold and 10-fold splits to find the optimal maximum depth of the tree. The Scikit-learn library supported three splitting criteria: "gini", "log_loss", and "entropy" (19). We ran the experiments with different combinations of these three splitting criteria and the optimal maximum depth from cross-validation to compare model performance (**Table 4, Figure 3**).

Neural networks can be used for both classification and regression models. They are made up of a series of layers with the first layer having many nodes that each represent an input, the last layer representing the output, and several hidden layers in between. Each hidden layer is made of many neurons (11, 13).

To compare how the different neural network architectures would impact the prediction of water potability, we designed the models with between 3 and 6 layers, with input layer shape as 9 and output layer shape as 1. The number of neurons per hidden layer was the multiple of 16 (**Table 5**).

We used the PyTorch Sequential module and Linear module to build the models and chose Rectified Linear Unit (ReLU) as the activation function between each layer. To calculate the loss, we used BCEWithLogitsLoss which internally combines the binary cross entropy loss function and Sigmoid layer into one layer (20). To minimize loss, we used the Adaptive Moment Estimation (Adam) optimizer with a learning rate of 0.00001. We trained the dataset for 500 epochs with a 128 batch-size per each epoch.

Received: April 11, 2024
Accepted: September 5, 2024
Published: March 31, 2025

REFERENCES

1. WHO, UNICEF, World Bank. "State of the World's Drinking Water: An urgent call to action to accelerate progress on ensuring safe drinking water for all." *Geneva: World Health Organization*, 2022.
2. Zhu, Mengyuan, et al. "A Review of the Application of Machine Learning in Water Quality Evaluation." *Eco-Environment & Health*, vol. 1, no. 2, Jun. 2022, pp. 107–116, https://doi.org/10.1016/j.eehl.2022.06.001.
3. WHO, UNICEF, World Bank. "Guidelines for drinking-water quality: fourth edition incorporating the first and second addenda." *Geneva: World Health Organization*, 2022.
4. Cojbasic, Sanja, et al. "Application of Machine Learning in River Water Quality Management: A Review." *Water Science & Technology*, vol. 88, no. 9, 13 Oct. 2023, pp. 2297–2308, https://doi.org/10.2166/wst.2023.331.
5. Haghiabi, Amir Hamzeh, et al. "Water Quality Prediction Using Machine Learning Methods." *Water Quality Research Journal*, vol. 53, no. 1, 19 Jan. 2018, pp. 3–13, https://doi.org/10.2166/wqrj.2018.025.
6. Nasir, Nida, et al. "Water Quality Classification Using Machine Learning Algorithms." *Journal of Water Process Engineering*, vol. 48, Aug. 2022, pp. 102920, https://doi.org/10.1016/j.jwpe.2022.102920.
7. Hassan, N, and C S Woo. "Machine Learning Application in Water Quality Using Satellite Data." *IOP Conference Series: Earth and Environmental Science*, vol. 842, no. 1, 1 Aug. 2021, pp. 012018, https://doi.org/10.1088/1755-1315/842/1/012018.
8. Wang, Xiaoping, et al. "Evaluation of Water Quality Based on a Machine Learning Algorithm and Water Quality Index for the Ebinur Lake Watershed, China." *Scientific Reports*, vol. 7, no. 1, 9 Oct. 2017, https://doi.org/10.1038/s41598-017-12853-y.
9. Nasir, Nida, et al. "Water Quality Classification Using Machine Learning Algorithms." *Journal of Water Process Engineering*, vol. 48, Aug. 2022, pp. 102920, https://doi.org/10.1016/j.jwpe.2022.102920.
10. Ubah, J. I., et al. "Forecasting Water Quality Parameters Using Artificial Neural Network for Irrigation Purposes." *Scientific Reports*, vol. 11, no. 1, 24 Dec. 2021, https://doi.org/10.1038/s41598-021-04062-5.
11. Goodfellow, Ian, et al. "Deep Learning." *MIT Press*, 2016.
12. Winterfeldt, Detlof Von and Ward Edwards. "Decision Analysis and Behavioral Research." *Cambridge University Press*, 1986, pp. 63–89.
13. Maimon, Oded and Lior Rokach. "Data Mining and Knowledge Discovery Handbook." *Springer US*, 2010.
14. Fix, Evelyn and J. L. Hodges. "Discriminatory Analysis: Nonparametric Discrimination, Consistency Properties." *USAF School of Aviation Medicine*, 1951.
15. Wang, Yisen and Shu-Tao Xia. "Unifying Attribute Splitting Criteria of Decision Trees by Tsallis Entropy." *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, https://doi.org/10.1109/ICASSP.2017.7952608.

https://doi.org/10.59720/24-143

16. Kadiwal, Aditya. “Water Quality.” *Kaggle,* 2021. www.kaggle.com/datasets/adityakadiwal/water-potability/data. Accessed 18 Feb. 2024.
17. “Cross Validation in Machine Learning.” *GeeksforGeeks*, 2023, www.geeksforgeeks.org/cross-validation-machine-learning/. Accessed 28 Feb. 2024.
18. Van Rossum, Guido and Fred Drake. “Python 3 Reference Manual.” *Python Software Foundation*, 2009. docs.python.org/3/reference/index.html. Accessed 15 Feb. 2024.
19. Pedregosa, Fabian, et al. “Scikit-Learn: Machine Learning in Python.” *ArXiv.org*, 5 Jun. 2018, arxiv.org/abs/1201.0490.
20. Ansel, Jason, et al. “PyTorch 2: Faster Machine Learning through Dynamic Python Bytecode Transformation and Graph Compilation.” *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, vol 2, Apr. 2024, pp. 929–947, https://doi.org/10.1145/3620665.3640366.
21. “Disinfectants and Disinfection Byproducts Rules: What Do They Mean to You?” *United States Environmental Protection Agency*, 2020. www.epa.gov/dwreginfo/dbprs-what-does-it-mean-you. Accessed 10 Apr. 2024.