Efficient Water Quality Analysis and Prediction using Machine Learning

Description

Water is considered as a vital resource that affects various aspects of human health and lives. The quality of water is a major concern for people living in urban areas. The quality of water serves as a powerful environmental determinant and a foundation for the prevention and control of waterborne diseases. However predicting the urban water quality is a challenging task since the water quality varies in urban spaces non-linearly and depends on multiple factors, such as meteorology, water usage patterns, and land uses, so this project aims at building a Machine Learning (ML) model to Predict Water Quality by considering all water quality standard indicators.

Abstract

Water is an essential resource for human existence. In fact, more than 60% of the human body is made up of water. Our bodies consume water in every cell, in the different organisms and in the tissues. Hence, water allows stabilization of the body temperature and guarantees the normal functioning of the other bodily activities. Nevertheless, in recent years, water pollution has become a serious problem affecting water quality. Therefore, to design a model that predicts water quality is nowadays very important to control water pollution, as well as to alert users in case of poor quality detection. Motivated by these reasons, in this study, we take the advantages of machine learning algorithms to develop a model that is capable of predicting the water quality index and then the water quality class.The use of the multiple regression algorithms has proven to be important and effective in predicting the water quality index. In addition, the adoption of the artificial neural network provides the most highly efficient way to classify the water quality.
Water quality analysis is required mainly for monitoring purposes. Some importance of such assessment includes:

- To check whether the water quality is in compliance with the standards, and hence, suitable or not for the designated use.
- To monitor the efficiency of a system, working for water quality maintenance
- To check whether upgradation / change of an existing system is required and to decide what changes should take place
- To monitor whether water quality is in compliance with rules and regulations.

Water quality analysis is extremely necessary for Public Health (especially for drinking water)

Literature Review

This project work explores the methodologies that have been employed to help solve problems related to water quality. Typically, conventional lab analysis and statistical analysis are used in research to aid in determining water quality, while some analyses employ machine learning methodologies to assist in finding an optimised solution for the water quality problem.. When it comes to estimating water quality using machine learning, Shafi et al. [1] estimated water quality using classical machine learning algorithms namely, Support Vector Machines (SVM), Neural Networks (NN), Deep Neural Networks (Deep NN) and k Nearest Neighbours (kNN), with the highest accuracy of 93% with Deep NN. The estimated water quality in their work is based on only three parameters: turbidity, temperature and pH, which are tested according to World Health Organisation (WHO) standards. Using only three parameters and comparing them to standardised values is quite a limitation when predicting water quality. Ahmad et al. [2] employed single feed forward neural networks and a combination of multiple neural networks to Water 2019, 11, 2210 3 of 14 estimate the WQI. They used 25 water quality parameters as the input.

Using a combination of backward elimination and forward selection selective combination methods, they achieved an R2 and MSE of 0.9270, 0.9390 and 0.1200, 0.1158, respectively. The use of 25 parameters makes their solution a little immoderate in terms of an inexpensive real time system, given the price of the parameter sensors. Sakizadeh [3] predicted the WQI using 16 water quality parameters and ANN with Bayesian regularisation. His study yielded correlation coefficients between the observed and predicted values of 0.94 and 0.77, respectively. Abyaneh [4] predicted the chemical oxygen demand (COD) and the biochemical oxygen demand (BOD) using two conventional machine learning methodologies namely, ANN and multivariate linear regression. They used four parameters, namely pH, temperature, total suspended solids (TSS) and total suspended (TS) to predict the COD and BOD. Ali and Qamar [5] used the unsupervised technique of the average linkage (within groups) method of hierarchical clustering to classify samples into water quality classes.

However, they ignored the major parameters associated with WQI during the learning process and they did not use any standardised water quality index to evaluate their predictions. Gazzaz et al. [4] used ANN to predict the WQI with a model explaining almost 99.5% of variation in the data. They used 23 parameters to predict the WQI, which turns out to be quite expensive if one is to use it for an IoT system, given the prices of the sensors. Rankovic et al. [6] predicted the dissolved oxygen (DO) using a feedforward neural network (FNN).

They used 10 parameters to predict the DO, which again defeats the purpose if it has to be used for a real-time WQI estimation with an IoT system. Most of the research either employed manual lab analysis, not estimating the water quality index standard, or used too many parameters to be efficient enough. Ahmad et al. [2] employed single feed forward neural networks and a combination of multiple neural networks to

estimate the WQI. They used 25 water quality parameters as the input. Using a combination of backward elimination and forward selection selective combination methods, they achieved an R2 and MSE of 0.9270, 0.9390 and 0.1200, 0.1158, respectively. The use of 25 parameters makes their solution a little immoderate in terms of an inexpensive real time system, given the price of the parameter sensors. Sakizadeh [3] predicted the WQI using 16 water quality parameters and ANN with Bayesian regularisation. His study yielded correlation coefficients between the observed and predicted values of 0.94 and 0.77, respectively. Abyaneh [4] predicted the chemical oxygen demand (COD) and the biochemical oxygen demand (BOD) using two conventional machine learning methodologies namely, ANN and multivariate linear regression. They used four parameters, namely pH, temperature, total suspended solids (TSS) and total suspended (TS) to predict the COD and BOD. Ali and Qamar [5] used the unsupervised technique of the average linkage (within groups) method of hierarchical clustering to classify samples into water quality classes.

However, they ignored the major parameters associated with WQI during the learning process and they did not use any standardised water quality index to evaluate their predictions. Gazzaz et al. [7] used ANN to predict the WQI with a model explaining almost 99.5% of variation in the data. They used 23 parameters to predict the WQI, which turns out to be quite expensive if one is to use it for an IoT system, given the prices of the sensors. Rankovic et al. [6] predicted the dissolved oxygen (DO) using a feedforward neural network (FNN). They used 10 parameters to predict the DO, which again defeats the purpose if it has to be used for a real-time WQI estimation with an IoT system. Most of the research either employed manual lab analysis, not estimating the water quality index standard, or used too many parameters to be efficient enough.

Machine Learning Algorithms

Here both regression and classification algorithms both. The regression algorithm is used to estimate the WQI and the classification algorithms to classify samples into the previously defined WQC. Here eight regression algorithms and 10 classification algorithms are used.
The following algorithms were employed in our study:

(1) Multiple Linear Regression Multiple linear regression is a form of linear regression used when there is more than one predicting variable at play. When there are multiple input variables, we use multiple linear regression to assess the input of each variable that affects the output, as reflected in Equation (3), where y is the output for which machine learning has been applied to predict the value, x is the observed value, β is the slope on the observed value, and $\in$ is the error term [8]. $y = \beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta kxk + \in$ (3)

(2) Polynomial Regression Polynomial regression is used when the relation between input and output variables is not linear and a little complex. We used a

higher order of variables to capture the relation of input and output variables, which is not as linear. We used the order of two. Using a higher order of variables does carry Water 2019, 11, 2210 8 of 14 the risk of overfitting, as reflected in Equation (4), where y is the output for which machine learning has been applied to predict the value, x is the observed value, β is the fitting value, i is the number of parameters considered, k is the order of the polynomial equation, and ∈i is the error term or residuals of the ith predictor [9]. We used it with 2-degree polynomials with an order of C. $y = \beta 0 + \beta 1xi + \beta 2x\ 2\ i + \beta 3x\ 3\ i + \ldots + \beta kx\ k\ i + \in i$, for i = 1, 2, . . . , n (4)

(3) Random Forest Random forest is a model that uses multiple base models on subsets of the given data and makes decisions based on all the models. In random forest, the base model is a decision tree, carrying all the pros of a decision tree with the additional efficiency of using multiple models [10].

(4) Gradient Boosting Algorithm This is the most contemporary algorithm used in most competitions. It uses an additive model that allows for optimization of differentiable loss function. We used it with a loss function of 'ls', a min_samples_split of 2 and a learning rate of 0.1 [11].

(5) Support Vector Machines Support vector machines (SVMs) are mostly used for classification but they can be used for regression as well. Visualising data points plotted on a plane, SVMs define a hyperplane between the classes and extend the margin in order to maximise the distinction between two classes, which results in fewer close miscalculations [12].

(6) Ridge Regression Ridge regression works on the same principles as linear regression, it just adds a certain bias to negate the effect of large variances and to void the requirement of unbiased estimators. It penalises the coefficients that are far from zero and minimises the sum of squared residuals [13,14].

(7) Lasso Regression Lasso regression works on the same principles as ridge regression, the only difference is how they penalise their coefficients that are off. Lasso penalises the sum of absolute errors instead of the sum of squared coefficients [15].

(8) Elastic Net Regression Elastic net regression combines the best of both ridge and lasso regression. It combines the method of penalties of both methods and minimises the loss function [16].

(9) Neural Net/Multi-Layer Perceptrons (MLP) Neural nets are loosely based on the structure of neurons. They contain multiple layers with interconnected nodes. They contain an input layer and output layer, and hidden layers in between these two mandatory layers. The input layer takes in the predicting parameters and the output layer shows the prediction based on the input. They iterate through each training data point and generalise the model by giving and updating the weight on each node of each layer. The trained model then uses

those weights to decide what units to activate based on the input. Multi-layer perceptron (MLP) is a conventional model of neural net, which is mostly used for classification, but it can be used for regression as well [17]. We used it for classification with the configuration of (3, 7) running for a maximum of 200 epochs using 'lbfgs' solver. Water 2019, 11, 2210 9 of 14

(10) Gaussian Naïve Bayes Naïve Bayes is a simple and a fast algorithm that works on the principle of Bayes theorem with the assumption that the probability of the presence of one feature is unrelated to the probability of the presence of the other feature [18].

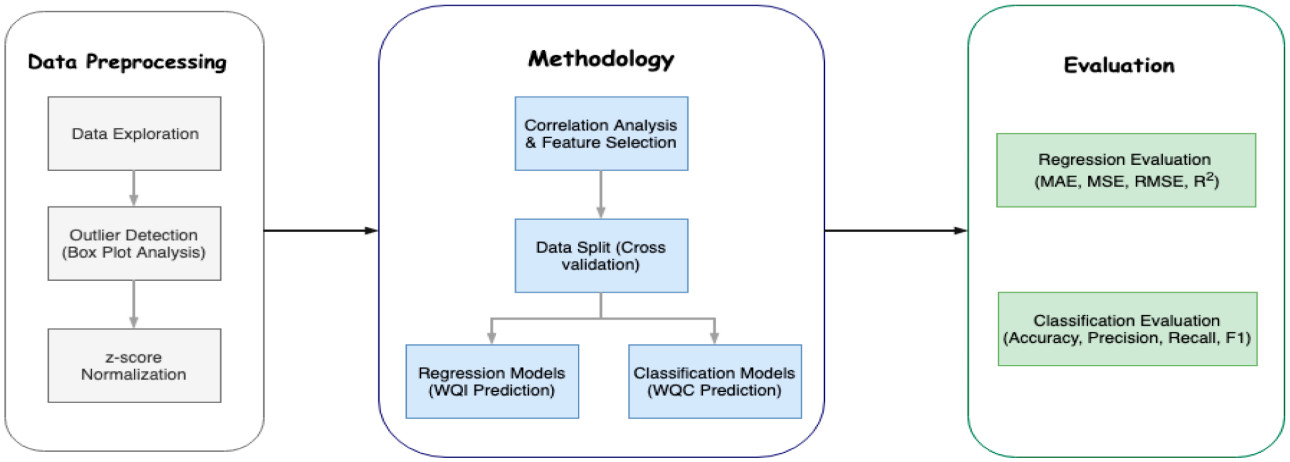
(11) Logistic Regression Logistic regression is a classification algorithm. It is based on the logistic function or the sigmoid function, hence the name. It is the most common algorithm used in the case of binary classification, but in our case we used multinomial logistic regression because there was more than two classes [19]. Stochastic gradient descent This iterative optimization algorithm minimizes the loss function iteratively to find the global optimum. In stochastic gradient descent, the sample selection is random [20].

(12) K Nearest Neighbor The KNN algorithm classifies by finding the given points nearest N neighbors and assigns the class of majority of n neighbors to it. In the case of a draw, one could employ different techniques to resolve it, e.g., increase n or add bias towards one class. K nearest neighbor is not recommended for large datasets [21]. We used a n = 5 configuration for our model.

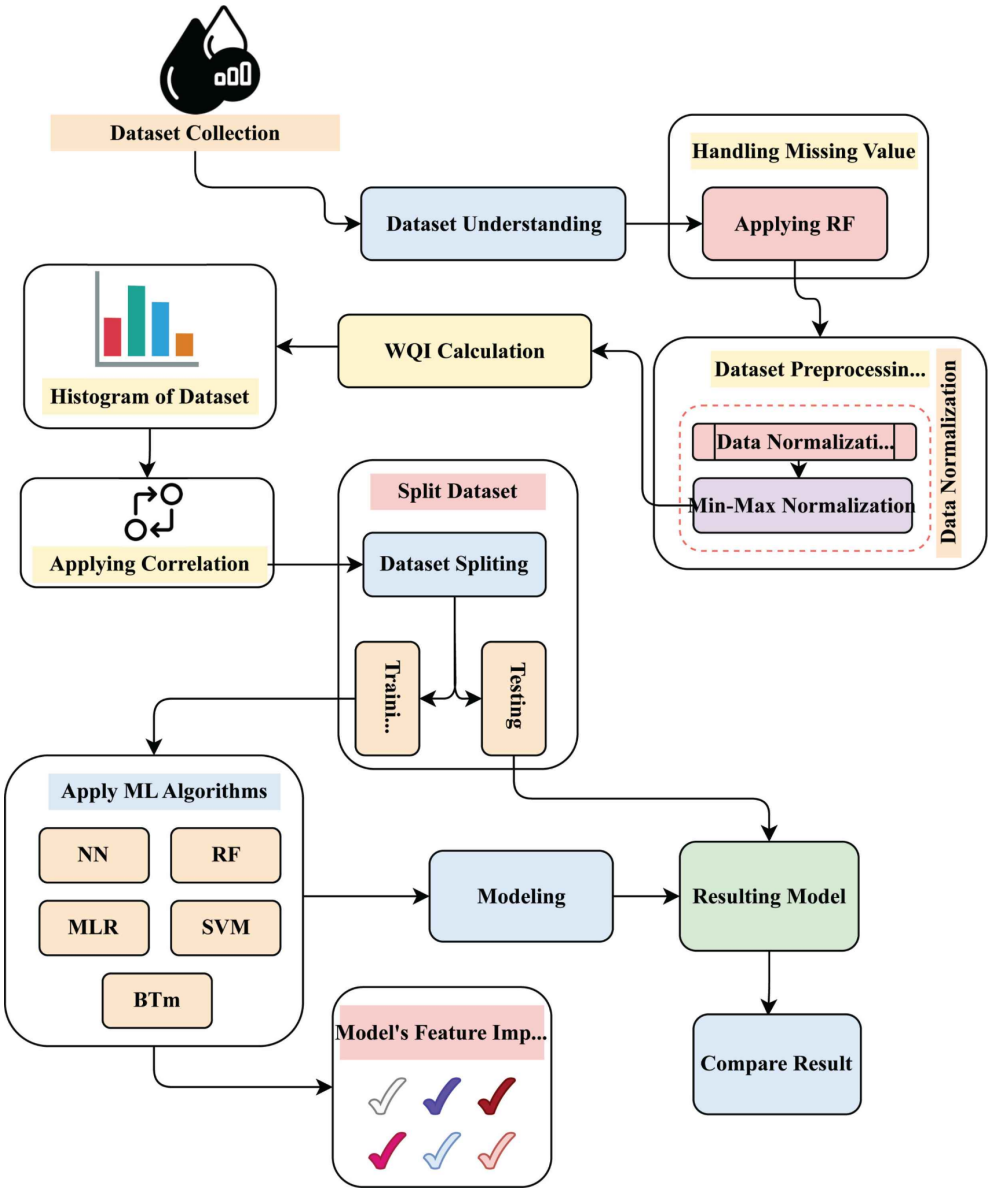
(13) Decision Tree A decision tree is a simple self-explanatory algorithm, which can be used for both classification and regression. The decision tree, after training, makes decisions based on values of all the relevant input parameters. It uses entropy to select the root variable, and, based on this, it looks towards the other parameters' values. It has all the parameter decisions arranged in a top-to-down tree and projects the decision based on different values of different parameters [22].

(14) Bagging Classifier A bagging classifier fits multiple base classifiers on random subsets of data and then averages out their predictions to form the final prediction. It greatly helps out with the variance [23]. We used default values for the algorithms, except MLP, which uses a (3, 7) configuration.

Diagrams:



Flowchart:



Reference:

https://careereducation.smartinternz.com/Student/guided_project_info/36147#

https://link.springer.com/article/10.1007/s40808-021-01266-6

https://www.mdpi.com/2073-4441/11/11/2210

Reference Publications:

1. Shafi, U.; Mumtaz, R.; Anwar, H.; Qamar, A.M.; Khurshid, H. Surface Water Pollution Detection using Internet of Things. In Proceedings of the 2018 15th International Conference on Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), Islamabad, Pakistan, 8–10 October 2018; pp. 92–96.
2. Ahmad, Z.; Rahim, N.; Bahadori, A.; Zhang, J. Improving water quality index prediction in Perak River basin Malaysia through a combination of multiple neural networks. Int. J. River Basin Manag. 2017, 15, 79–87.
3. Sakizadeh, M. Artificial intelligence for the prediction of water quality index in groundwater systems. Model. Earth Syst. Environ. 2016, 2, 8.
4. Abyaneh, H.Z. Evaluation of multivariate linear regression and artificial neural networks in prediction of water quality parameters. J. Environ. Health Sci. Eng. 2014, 12, 40.
5. Ali, M.; Qamar, A.M. Data analysis, quality indexing and prediction of water quality for the management of rawal watershed in Pakistan. In Proceedings of the Eighth International Conference on Digital Information Management (ICDIM 2013), Islamabad, Pakistan, 10–12 September 2013; pp. 108–113.
6. Rankovi´c, V.; Radulovi´c, J.; Radojevi´c, I.; Ostoji´c, A.; Comi´c, L. Neural network modeling of dissolved oxygen ˇ in the Gruža reservoir, Serbia. Ecol. Model. 2010, 221, 1239–1244.
7. Gazzaz, N.M.; Yusoff, M.K.; Aris, A.Z.; Juahir, H.; Ramli, M.F. Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors. Mar. Pollut. Bull. 2012, 64, 2409–2420.
8. Amral, N.; Ozveren, C.; King, D. Short term load forecasting using multiple linear regression. In Proceedings of the 2007 42 nd International Universities Power Engineering Conference, Brighton, UK, 4–6 September 2007; pp. 1192–1198.
9. Ostertagová, E. Modelling using polynomial regression. Procedia Eng. 2012, 48, 500–506.
10. Liaw, A.; Wiener, M. Classification and regression by randomForest. R News 2002, 2, 18–22.
11. Friedman, J.H. Stochastic gradient boosting. Comput. Stat. Data Anal. 2002, 38, 367–378.
12. Tong, S.; Koller, D. Support vector machine active learning with applications to text classification. J. Mach.Learn. Res. 2001, 2, 45–66.

13. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 1970, 12, 55–67.
14. Zhang, Y.; Duchi, J.; Wainwright, M. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. J. Mach. Learn. Res. 2015, 16, 3299–3340.
15. Tibshirani, R. Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B 1996, 58, 267–288.
16. Zou, H.; Hastie, T. Regression shrinkage and selection via the elastic net, with applications to microarrays.J. R. Stat. Soc. Ser. B 2003, 67, 301–320.
17. Günther, F.; Fritsch, S. Neuralnet: Training of neural networks. R J. 2010, 2, 30–38.
18. Zhang, H. The optimality of naive Bayes. AA 2004, 1, 3.
19. Hosmer, D.W., Jr.; Lemeshow, S.; Sturdivant, R.X. Applied Logistic Regression; John Wiley Sons: Hoboken, NJ, USA, 2013.
20. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010, Paris, France, 22–27August 2010; pp. 177–186.
21. Beyer, K.; Goldstein, J.; Ramakrishnan, R.; Shaft, U. When is "nearest neighbor" meaningful? In Proceedings of the International Conference on Database Theory, Jerusalem, Israel, 10–12 January 1999; pp. 217–235.
22. Quinlan, J.R. Decision trees and decision-making. IEEE Trans. Syst. Man Cybern. 1990, 20, 339–346.
23. Breiman, L. Bagging predictors. Mach. Learn. 1996, 24, 123–140.