CS-C3240
Federated Learning

Predicting Water Potability: A Comparison of Logistic Regression and Decision Tree Classifier

Arun Bhatia & Jesse Lukkarinen
Aalto University, Espoo, Finland

Contents

1. ABSTRACT

Water is essential for any living organism, but the quality of water could be non-potable and hence not healthy or safe for human consumption. High water quality measurements are essential for providing information on where drinking water is safe to drink without risk of potential illness or harmful substances. Federated learning is a machine learning method answering to privacy issues while sharing data in distributed computing and enabling collaborative model training. Predicting water quality can be enhanced with federated learning by allowing different locations with different amounts of data to collaboratively train models without sharing raw data. Water potability prediction federated stochastic gradient descent is used as the federated learning algorithm and local models are compared: logistic regression and decision tree classifier. Decision tree classifier performed better than logistic regression in both average prediction accuracy and smaller error accumulation, though there are signs of overfitting with decision tree classifier.

Index terms - Federated Learning, Networks, Classification, Logistic regression, Decision Tree Classifier, Logistic Loss, Machine Learning, Water Potability

2. INTRODUCTION

Water is a crucial resource for human survival, and it is a fundamental human right to have access to safe drinking water. However, not all water sources are safe for consumption, and contamination of water can lead to various waterborne diseases, such as cholera, typhoid, and hepatitis A, which can result in severe illness and even death [1]. As a result, it is critical to ensure that the water we drink is potable, which means that it is free from harmful substances that can cause health problems.

The World Health Organization (WHO) has designated guidelines for the quality of drinking water and recommends that the water we consume should adhere to specific standards [2]. Potable water should be free from harmful microorganisms, chemicals, and radiological hazards. Additionally, it should have a balanced pH, proper mineral content, and be free from any turbidity or odour. Water that does not meet these standards can pose a risk to human health.

Machine learning algorithms can analyze large datasets generated from water quality testing to identify patterns and correlations that may not be immediately evident to humans. By applying machine learning to water quality data, we can predict the potability of water samples and determine the factors that impact water quality. In this paper, we train machine learning models to classify water samples as potable or non-potable based on their physicochemical properties. These models can also identify the most significant variables that contribute to water quality, such as pH, turbidity, and mineral content.

3. RELEVANT LITERATURE

Federated learning is a machine learning technique utilized for preserving privacy and enhancing training performance without sharing vital data like in distributed computing. Federated learning allows several users or clients to train models collaboratively. This training does not happen by sharing valuable data to centralized, but by sharing for example gradient information acquired by each node in federated graph training a local model. [3,4] Quality data for training models is one of the main challenges for deep learning-based machine learning applications such as defect detection by computer vision, but federated learning allows data privacy and training models requiring large amounts of data by extracting certain features for training. [5] Similar research to this report has been completed by authors of [6], who utilized federated learning to enhance the model's classification performance in water quality prediction. With their Federated averaging implementation, they could increase the f1 score from the range of 0.4-0.5 to 0.6-0.7. According to their results, federated learning approaches improved the performance of classification methods past more sophisticated methods. [6] Federated learning has been used previously for water quality prediction and in this report, two methods are compared for water quality prediction.

Section 4 of this paper is about problem formulation. It contains information about data points, features, labels and type data. Section 5 discusses about the methods that were used. In section 6, the results are conveyed and compared. Section 7 has a further analysis of the results and a discussion of what could have differently. The last chapter, section 8 is about the references.

4. PROBLEM FORMULATION

The potability of water is affected by multiple variables. For the problem of classifying water potability, the label variable is potability and feature variables are other variables in the dataset, which are: Ph, hardness, solids, chloramines, sulfate, conductivity, organic_carbon, trihalomethanes and turbidity.

The label variable (potability) gets only two binary values 0 or 1. 0 means that the water is not safe for humans to drink. On the other hand, 1 indicates that water is safe for human consumption. The other 9 features all get real values of 0 or greater than 0. All in all, a data point in the dataset contains one label and nine feature values as described above.

In an empirical graph, the edges serve as representations of the connections or relationships that exist between various nodes or entities within the graph. These edges are accompanied by weights, which play a crucial role in quantifying the strength or significance of these relationships. In section 3, we will go more deeply through the process of adding local datasets and models to each node. Furthermore, the weights and edges are also chosen later.

The dataset that was used in this project was from Kaggle. It is freely available there for anyone to use it. https://www.kaggle.com/datasets/adityakadiwal/water-potability. The dataset composes of 3276 water quality metrics of various water bodies.

5. METHODS

The original dataset composes of 3276 water quality metrics of various water bodies. The local datasets were made of the original dataset by splitting the original dataset into 5 local datasets based on the pH values. Below are the range pH of values and the number of data points that each local dataset contains. Ph levels can be classified to Soft 0 to 17.1, Slightly Hard 17.1 to 60, Moderately Hard 60 to 120, hard >120 to 180 Very Hard > 180. [7]

Range of pH: (5.6, 8.4], Number of data points: 2304
Range of pH: (8.4, 11.2], Number of data points: 498
Range of pH: (2.8, 5.6], Number of data points: 437
Range of pH: (11.2, 14.0], Number of data points: 21
Range of pH: (-0.014, 2.8], Number of data points: 16

As we can see pH values of water bodies between 5.6 and 8.4 have the highest occurrence in the dataset. The last range, which is between -0.014 and 2.8 should be between 0 and 2.8 but there seems to be some error with compilers dividing small numbers.

Other features are hardness, solids, sulfate, conductivity, organic carbon, trihalomethanes, and turbidity, which all have an effect on the quality of water. Water hardness can be measured as the total calcium and magnesium concentration in a sample of water [8]. Solids in the data are dissolved solids, which can be either organic or inorganic and is a good measure of water suitability for drinking [9]. Chloramines are substances used for disinfecting water systems. Sulfates are substances that are typically found in different mineral compounds and soil. The conductivity of water is a good indicator of water quality as pure water does not conduct electricity well. Organic carbon levels of water depend on decaying material in water. Turbidity represents the amount of solid matter in water. [10] Consuming high levels of trihalomethanes over a long period of time might increase risk of bladder cancer and other health risks [11].

The correlation matrix plotted in the notebook helps us to get insights into the relationships between features. Meaning we can identify which features may be relevant to our application. Based on the correlation matrix we think that all 9 features should be used. We understand that the features with high negative or positive correlation will be the most useful in prediction of the water potability. We also think that the 9 features each have almost the same effect on the potability of the water, so we should not drop any of the features.

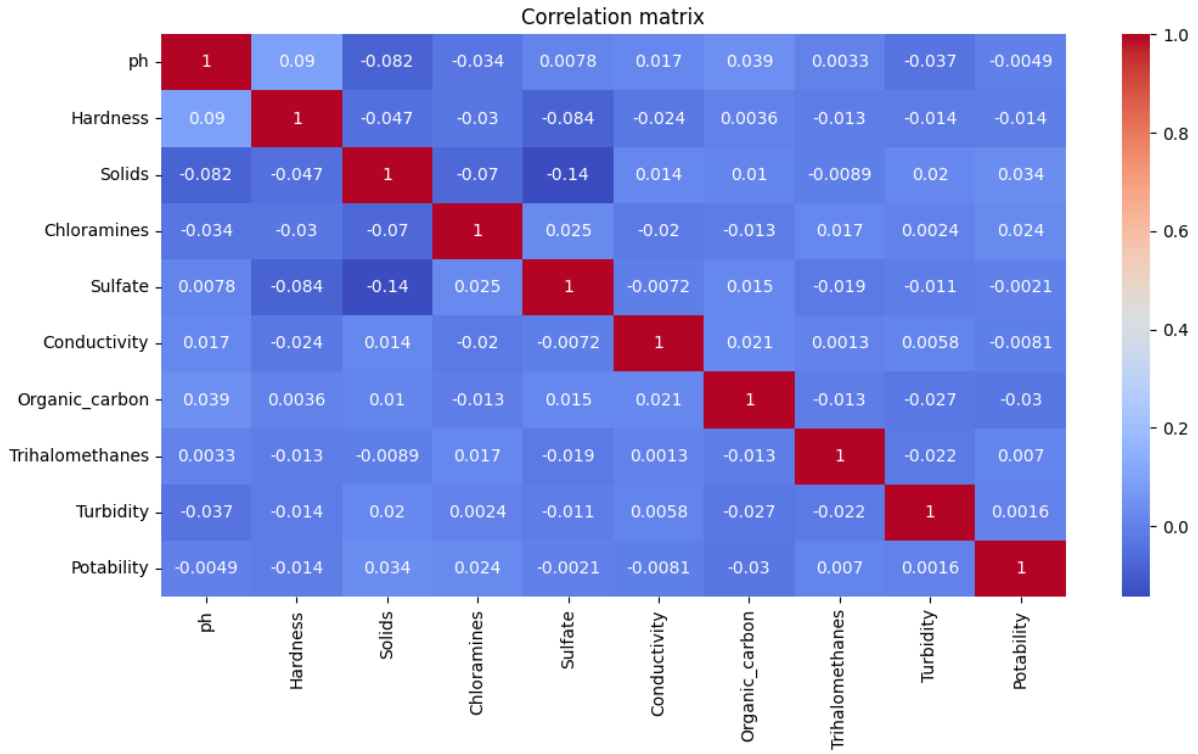

Figure 1. Water potability correlation matrix

The local datasets were then divided into training, testing and validation datasets using sklearn's train_test_split function. The testing set consists of 60% of the data points in each local dataset. The training and validation set each consist of 20% of the data points in each local dataset. The number of data points in each of the 5 local datasets split into training, testing and validation datasets are below. As we can see, the validation and testing local datasets contain roughly the same number of data points, while the training set contains a larger amount of data points than the two previous ones.

PH Range: (-0.014, 2.8]
Training set datapoints: 9, Testing set datapoints: 3 Validation set datapoints: 4

PH Range: (2.8, 5.6]
Training set datapoints: 262, Testing set datapoints: 87 Validation set datapoints: 88

PH Range: (5.6, 8.4]
Training set datapoints: 1382, Testing set datapoints: 461 Validation set datapoints: 461

PH Range: (8.4, 11.2]
Training set datapoints: 298, Testing set datapoints: 100 Validation set datapoints: 100

PH Range: (11.2, 14.0]
Training set datapoints: 12, Testing set datapoints: 4 Validation set datapoints: 5

The network graph was constructed using sklearn's networkx function. Each node in the graph carries a local dataset and local model. The local dataset is made of two datasets x_train and y_train. X_train has all the 9 features and y_train has the labels values. This means the y_train

dataset has categorical values of the potability. In the notebook, we have plotted 5 plots of the y_train scatter plots. By looking at the plots we decided which nodes were most similar to each other. In the end, we connected the nodes that were connected are (1 and 2) (0 and 3) and (4 and 2). For all nodes we used logistic loss as the loss function to learn the hypothesis.

The two different federated learning methods based on GTV minimization that are using the same federated learning optimization algorithm FedSGD (federated stochastic gradient descent) and the same choices for the local models are logistic regression and decision tree classifier. The loss function for both methods is the logistic loss which is sometimes called also the cross-entropy loss. Using logistic loss is motivated as logistic regression commonly used log loss as loss function. Also log loss is used in classification models and classification of water potability is our main target.

Firstly, we chose logistic regression because it is well-suited for applications where the variable gets categorical and binary values. Furthermore, logistic regression is relatively robust to outliers compared to other calcification techniques [13].

Secondly, we chose a decision tree classifier since decision trees are easily interpretable models. The tree structure represents a series of hierarchical decision rules, where each leaf node represents a predicted class. Furthermore, decision trees have efficient training and prediction times, making them suitable for large datasets [14].

Because we split the original dataset into 5 sub-datasets, we will have a local model for each node in the two federated learning methods. Each node will also have a local loss function. For nodes 0-4 in the logistic regression, the logistic loss function has been used to train the model. We used the logistic loss because we have a binary classification task (classification of water into potable and non-potable). Meaning the label values of the local datasets have binary values. Logistic loss measures the inconsistency between the predicted probabilities and the true labels of the class.

We also opted to use the logistic loss function for nodes 0-4 in the decision tree classifier. It is not common to use the logistic loss with decision trees but it is still since it is appropriate for probabilistic models like DecisionTreeClassifier. Again the label values of the local datasets have binary values.

All in all, we obtain two different FL methods by using the same FL (optimization) algorithm (FedSGD) and the same choice for loss function but a different classification method.

To measure the variation of local models we decided to compare the values average testing error of both logistic regression and decision tree classifier. The average testing error was calculated on the common test set which contained about 20% of all the data points. The average testing error was implemented using Sklearn's accuracy_score metric which computes the accurateness of a set of predicted labels against the actual true labels.

As mentioned before. Sklearn's train_test_split function was used for splitting each local dataset into training, validation and testing sets. It was a single split where the training set contained 60% of the local datapoints, the testing set contained 20% of the local datapoints and the validation set also contained 20% of the local datapoints.

6. RESULTS

Results may vary on depending on how data is split into training, testing and validation datasets as each run of the whole program will change the data in each dataset.

Logistic regression and logistic loss were used as the first method to predict water potability.

Node	Log Loss Error
1 Train Validation	 24.02910225941143 27.032740041837865
2 Train Validation	 24.07495932479199 24.165631249521724
3 Train Validation	 21.464491128251385 20.797422562917916
4 Train Validation	 12.458041272077406 11.17353255062632
5 Train Validation	 30.036377824264292 21.626192033470293

Logistic regression/Logistic loss	Value
Training accuracy average	0.378181947323729
Validation accuracy average	0.4185077893906527
Average Training Log Loss	17.86093829373976
Average Validation Log Loss	20.959103687674823

Decision tree classifier was used as the second local model and with loss function logistic loss.

Node	Log Loss Error

1 Train Validation	 12.014551129705717 27.032740041837865
2 Train Validation	 2.2204460492503136e-16 12.287609109926303
3 Train Validation	 2.220446049250314e-16 11.962427263633243
4 Train Validation	 11.490426415993724 16.219644025102717
5 Train Validation	 15.018188912132146 7.20873067782343

Decision tree classifier/Logistic loss	Value
Training accuracy average	0.786241610738255
Validation accuracy average	0.5854407414711102
Average Training Log Loss	7.704633291566317
Average Validation Log Loss	14.942230223664712

7. CONCLUSION

By looking at the training and validation errors for the logistic regression and decision tree classifier models we think that the decision tree classifier is performing better and it will be the final chosen model. The average logistic loss values of the decision tree classifier in the training and validation sets are lower compared to the logistic regression. Furthermore, the training and validation accuracy averages are higher in the decision tree classifier, which means that it's performing better. We also think that there might be some overfitting with the decision tree classifier but its values for average logistic loss and average accuracy score are much better than logistic regression's.

Because we have split the local models into three sets, we already have a test set that consists of data points that have neither been used to train the local models (training set) nor for choosing between different local models (validation set).

We calculate the testing average logistic loss and testing average accuracy score for the decision tree classifier model. The testing accuracy average is close to the validation accuracy

average and the average testing log loss is close to the validation testing log loss. Because of these values, we think that our decision tree classifier model is working well.

Decision tree classifier/Logistic loss	Value
Testing accuracy average	0.5307278529932431
Average Testing Log Loss	16.914282611878377

We also confirmed that we made the right choice in choosing the decision tree classifier model by calculating the average testing errors for the logistic regression model also. The errors for each node can be found in the notebook but the testing accuracy average is larger in the testing of the decision tree classifier. In addition, the average testing log loss is smaller in the decision tree classifier. This confirms that on a random set where the samples are in the local training set, the decision tree classifier will work better.

Logistic regression/Logistic loss	Value
Testing accuracy average	0.409426284688458
Average Testing Log Loss	21.2864342954123

We also plotted the average errors that were computed. Below are the plots for the average logistic loss and average accuracy scores.

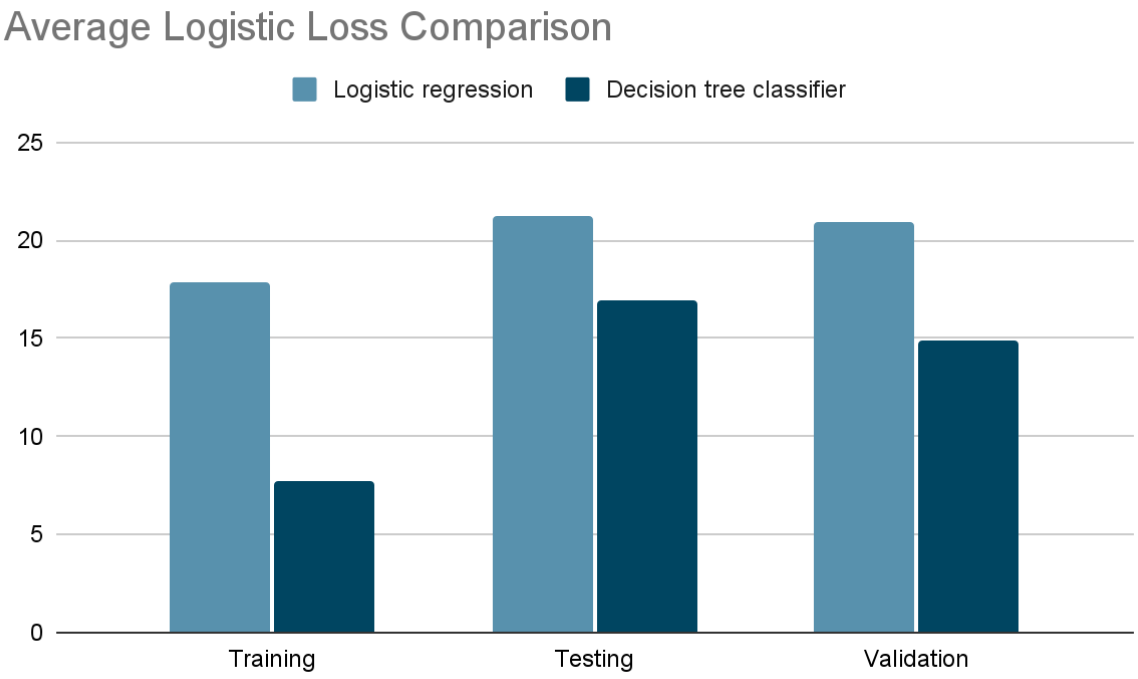

Figure 2. Average logistic loss score comparison

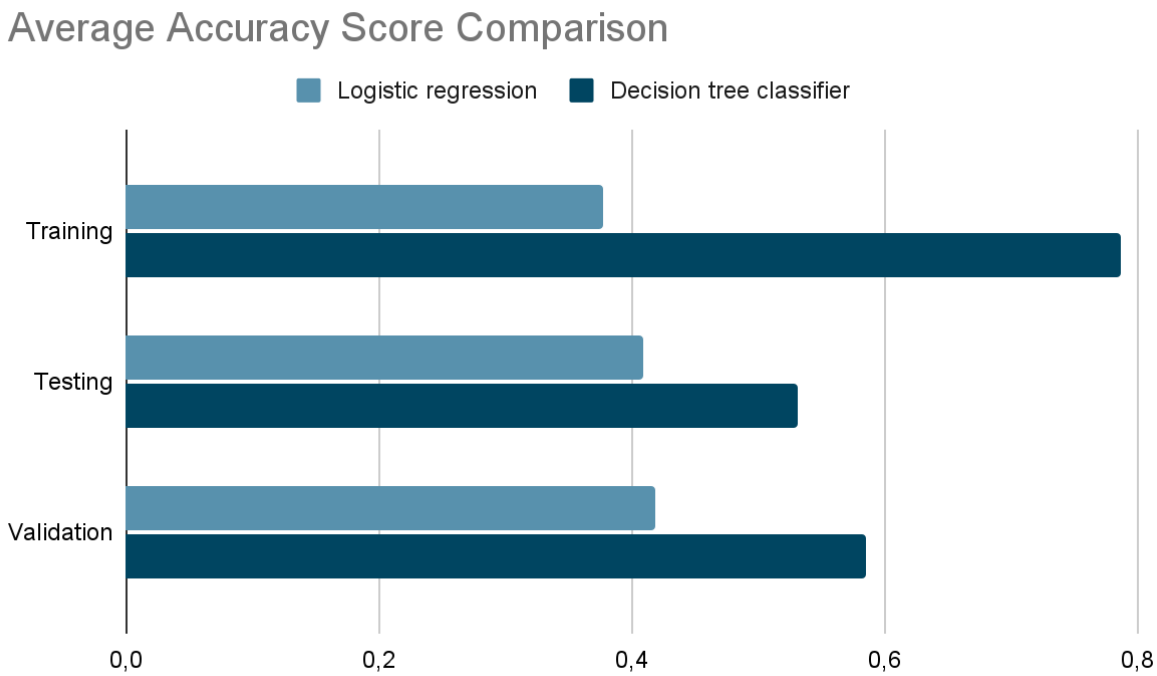

Figure 3. Average accuracy score comparison

For future work, we think that the dataset needs to be larger and contain more values high and low values of the pH. For example, the pH range: (-0.014, 2.8] contains only 9 training datapoints and the pH range: (11.2, 14.0] contains only 12 training datapoints. Perhaps it might have been better to split the pH ranges according to the number of datapoints in each range. This way each split would have almost the same number of datapoints (3276 / 5 ≈ 655 datapoints) but the pH ranges would not be the same length.

We also could have used similarity measures from statistics for choosing edges and weights for the network graph. This way, we might have connected the edges that actually needed to be connected rather than by looking at the plots.

The training error at some local nodes was much smaller than the validation error in the decision tree classifier model which hints at overfitting. The beginning of Chapter 5 had some discussion about overfitting due to differences in the average errors. For example, the training logistic loss for node 2 in the decision tree classifier was 2.2204460492503136e-16 and the validation logistic loss for the same node was 12.287609109926303. We can clearly see that the decision tree classifier model overfits here. The same thing also happened in node 3.

In future work, one way to fix the overfitting would be decision tree pruning [15]. Because we built the tree to achieve high purity there is a big change of overfitting that can be fixed with decision tree pruning. Possibly a better solution could be to use random forest models (random decision forests). Random forest models are clusters of decision trees and the majority vote of the forest is picked as the predicted output [15]. Random forests are also more robust and accurate compared with decision trees.

Finally, if we could be provided with exponentially more training data on the water data then using neural networks seems to be the best choice. Neural networks tend to perform better on larger amounts of data. So, the performance (average loss) would become more satisfactory with larger datasets and neural networks.

8. REFERENCES

[1] Disease Impact of Unsafe Water, https://www.cdc.gov/healthywater/global/disease-impact-of-unsafe-water.html, (Online: 08.05.2023)

[2] Guidelines for drinking-water quality, 4th edition, incorporating the 1st addendum, https://www.who.int/publications/i/item/9789241549950, (Online: 08.05.2023)

[3] L. Li, Y. Fan, M. Tse, and K. Y. Lin, “A review of applications in federated learning,” *Comput Ind Eng*, vol. 149, p. 106854, Nov. 2020, doi: 10.1016/J.CIE.2020.106854.

[4] T. Sun, D. Li and B. Wang, "Decentralized Federated Averaging," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 4, pp. 4289-4301, 1 April 2023, doi: 10.1109/TPAMI.2022.3196503.

[5] Han, X., Yu, H., Gu, H. (2019). Visual Inspection with Federated Learning. In: Karray, F., Campilho, A., Yu, A. (eds) Image Analysis and Recognition. ICIAR 2019. Lecture Notes in Computer Science(), vol 11663. Springer, Cham. https://doi.org/10.1007/978-3-030-27272-2_5

[6] V. J, K. K, G. M. P, G. C, P. R. Subramaniam, and S. Rangarajan, “Strategies for classifying water quality in the Cauvery River using a federated learning technique,” *International Journal of Cognitive Computing in Engineering*, vol. 4, pp. 187–193, Jun. 2023, doi: 10.1016/J.IJCCE.2023.04.004.

[7] Lehr, J. et. al., 1980. Domestic Water Treatment, New York, NY: McGraw-Hill Book Company

[8] K. M. Brown and C. Lydeard, “Mollusca: Gastropoda,” *Ecology and Classification of North American Freshwater Invertebrates*, pp. 277–306, Jan. 2010, doi: 10.1016/B978-0-12-374855-3.00010-8.

[9] *The Berkey.* What Is The Acceptable Total Dissolved Solids (TDS) Level In Drinking Water?. Available: https://theberkey.com/blogs/water-filter/what-is-theacceptable-total-dissolved-solids-tds-level-in-drinking-water

[10] Aditya Kadiwal. Water quality dataset. Available: https://www.kaggle.com/datasets/adityakadiwal/water-potability

[11] Iowa Public Health Tracking Portal. Public Drinking Water and Health. Available: https://tracking.idph.iowa.gov/Environment/Public-Drinking-Water/Public-Water-and-Health/TTHM-in-Public-Water-and-Health

[12] Amir Hamzeh Haghiabi; Ali Heidar Nasrolahi; Abbas Parsaie, Water quality prediction using machine learning methods, Water Quality Research Journal (2018) 53 (1): 3–13.

[13] Essential guide to handle Outliers for your Logistic Regression Model, https://medium.com/geekculture/essential-guide-to-handle-outliers-for-your-logistic-regression-model-63c97690a84d, (Online: 30.05.2023)

[14] A Guide to Decision Trees for Machine Learning and Data Science, https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956, (Online: 30.05.2023)

[15] Comparative Study on Classic Machine learning Algorithms, https://towardsdatascience.com/a-guide-to-decision-trees-for-machine-learning-and-data-science-fe2607241956, (Online: 30.05.2023)