WATER QUALITY PROJECT

Statistical Learning Module

Simona Caruso

January 7, 2024

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

Contents

1 Introduction

The following project was conducted using the Kaggle Water Quality dataset with the aim of understanding the interpretative differences among various models utilized. The selected models can be categorized into supervised and unsupervised models. In this analysis, priority was given to interpretability over predictive performance. Firstly, we will delve into exploratory analysis and data cleansing to comprehend the nature of the dataset. Subsequently, we will elucidate and showcase the outcomes derived from the different models.

2 Exploratory Data Analysis

The dataset under analysis comprises 7999 observations and 21 variables listed below:

- aluminium: dangerous if greater than 2.8
- ammonia: dangerous if greater than 32.5
- arsenic: dangerous if greater than 0.01
- barium: dangerous if greater than 2
- cadmium: dangerous if greater than 0.005
- chloramine: dangerous if greater than 4
- chromium: dangerous if greater than 0.1
- copper: dangerous if greater than 1.3
- fluoride: dangerous if greater than 1.5
- bacteria: dangerous if greater than 0
- viruses: dangerous if greater than 0
- lead: dangerous if greater than 0.015
- nitrates: dangerous if greater than 10
- nitrites: dangerous if greater than 1
- mercury: dangerous if greater than 0.002
- perchlorate: dangerous if greater than 56
- radium: dangerous if greater than 5
- selenium: dangerous if greater than 0.5
- silver: dangerous if greater than 0.1
- uranium: dangerous if greater than 0.3
- is_safe: class attribute 0 - not safe, 1 - safe

All variables considered are continuous numeric, except for the last variable 'is_safe', which is treated as factorial. However, the variable 'ammonia' was erroneously read as character and was therefore transformed into numeric. Additionally, the 'is_safe' variable has an extra level, #NUM!, beyond those previously indicated. Since this level is associated with only three observations, it was simply preferred to eliminate them. In addition, these three observations also had missing values (NA) for the ammonia variable. Furthermore, concerning the ammonia variable, some observations contained negative values, which are not feasible and were consequently removed.

Referring to the main statistics of the variables, peculiarities become apparent. Visual aids, such as a density plots, illustrate the nature of variables' distributions; meanwhile, the boxplots help us to detect outliers. Outliers are identified for the variables 'aluminium',

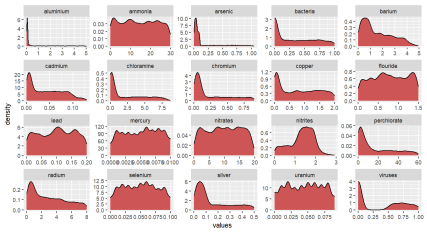

Figure 1: Histograms for variables' distributions

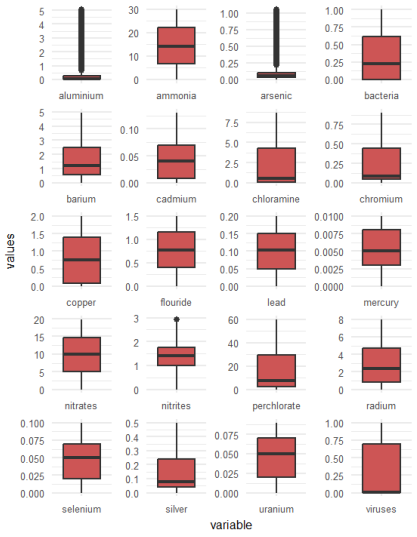

Figure 2: Boxplots for variables

'arsenic' and 'nitrites'. While these outliers might pose challenges during model creation, it seemed better not to eliminate or handle these observations differently as they don't significantly deviate from one another. The same rationale applies to the 'arsenic' variable.
The 'aluminium' variable exhibits a notably different maximum value compared to the third quartile, and it's observed that the mean surpasses the median. The 'viruses' variable displays a distinctive distribution: a substantial portion of observations cluster around zero, while a smaller subset falls between 0.5 and 1, potentially indicating water non-potability.

'Nitrites' present a thick left tail in their distribution, with a density peak occurring between values 1 and 2. Given that values exceeding 1 in this variable could signify danger, it will be intriguing to explore their impact on predictions.
Lastly, considering the response variable 'is_safe': 7076 water samples are classified as safe, whereas 910 are labeled as unsafe.

Through multivariate analysis, we can observe how variables interact with each other. From the correlation matrix, depicted in Figure 3, we notice a lack of significant correlation among the variables.

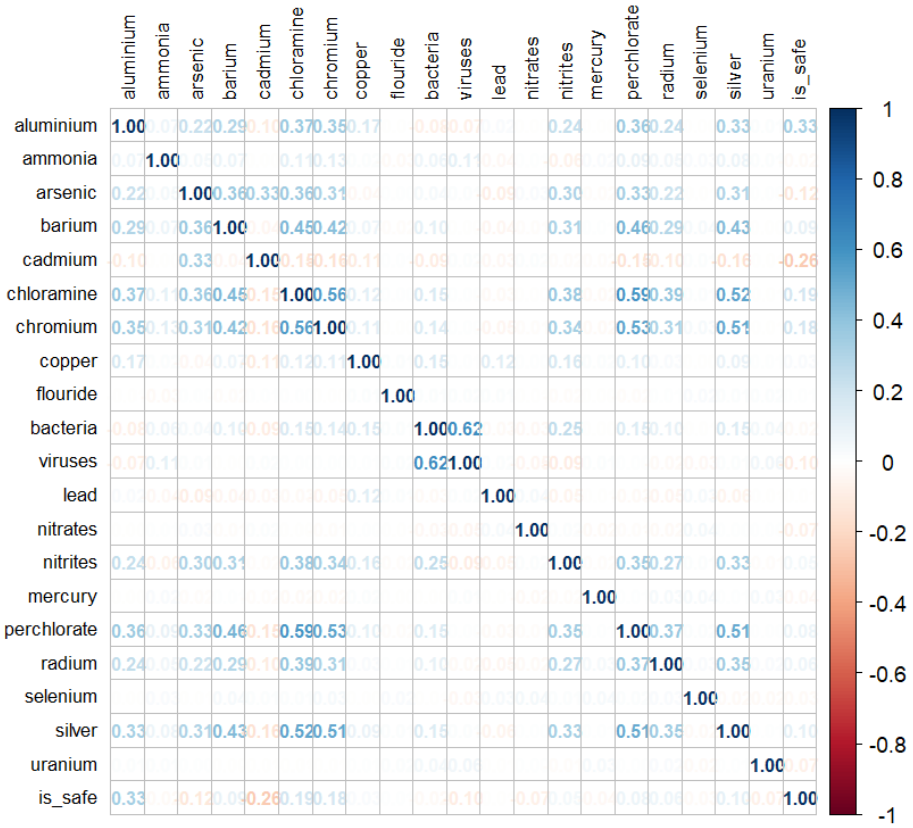

Figure 3: Correlation Matrix

3 Supervised Learning

For supervised learning models, the dataset was split into a training set, containing 80% of the data, and a test set with the remaining 20%. This division is crucial because during model creation, we aim to prevent excessive adaptation to the available data and ensure that the models perform well on unseen data. Additionally, some models require standardized data, as variations in the range of values across variables might impact the model's quality. For each model under analysis, a brief explanation will be provided along with comments on the obtained results.

The measures used to evaluate model performance are sensitivity, specificity, and balanced accuracy. Sensitivity represents the proportion of true positives identified by the model out of all positive cases, while specificity indicates the proportion of cases correctly identified as negative out of all negative instances. Since identifying both positive and negative cases is crucial for our objectives, we used balanced accuracy as a performance metric as well. Balanced accuracy is simply the average of these two measures described above.

3.1 Logistic Regression

Logistic regression is a statistical technique used to predict a binary dependent variable by employing a sigmoid function. The resulting prediction translates into a conditional probability, expressed as follows:

$$p(Y = 1|X) = \frac{e^{(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}}{1 + e^{(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}} \tag{1}$$

Here, $p(Y = 1|X)$ represents the conditional probability of Y taking the value of 1, given the set of independent variables $X_1, X_2, \ldots, X_n$, while $\beta_0, \beta_1, \ldots, \beta_n$ are the estimated coefficients.

To obtain accurate estimates of the coefficients associated with the variables, the likelihood function is maximized, although the specific details of this procedure are beyond the scope of this discussion. However, it is crucial to emphasize that the direct interpretation of these coefficients is intricate as they do not represent a linear measure of impact on probability.

To simplify the interpretation of coefficients, Equation (1) can be rewritten as follows:

$$\log\left(\frac{p(Y = 1|X)}{1 - p(Y = 1|X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{2}$$

The quantity on the left side of the equation is termed log-odds or logit, and now the coefficients can be interpreted as a direct measure of this quantity.

The model results are presented in Figure 4.

The coefficient estimates for all variables, except for fluoride, are significant within a 95% confidence interval. These estimates should be understood as the effects of a unit increase in the associated variable, while keeping other variables constant. Thus, an increase in the values of variables like aluminum, chloramine, chromium, and bacteria elevates the likelihood of $Y = 1$, indicating unsafe water. Conversely, an increase in variables such as ammonia, arsenic, cadmium, copper, viruses, lead, nitrates, nitrites, mercury, perchlorate, radium, selenium, silver, and uranium decreases this probability.

Characteristic	log(OR)[1]	95% CI[1]	p-value
aluminium	0.70	0.63, 0.77	<0.001
ammonia	-0.03	-0.04, -0.02	<0.001
arsenic	-3.4	-4.1, -2.7	<0.001
barium	0.12	0.03, 0.21	0.006
cadmium	-21	-25, -17	<0.001
chloramine	0.18	0.13, 0.22	<0.001
chromium	1.3	0.90, 1.7	<0.001
copper	-0.41	-0.56, -0.25	<0.001
flouride	0.16	-0.06, 0.37	0.2
bacteria	0.82	0.35, 1.3	<0.001
viruses	-1.4	-1.8, -1.0	<0.001
lead	-1.8	-3.5, -0.16	0.032
nitrates	-0.05	-0.07, -0.03	<0.001
nitrites	-0.32	-0.53, -0.10	0.004
mercury	-34	-64, -2.7	0.033
perchlorate	-0.02	-0.03, -0.02	<0.001
radium	-0.05	-0.09, 0.00	0.039
selenium	-4.8	-8.0, -1.6	0.004
silver	-1.3	-2.1, -0.54	<0.001
uranium	-13	-17, -9.8	<0.001

[1] OR = Odds Ratio, CI = Confidence Interval

Figure 4: Logistic Regression results

3.1.1 Performance Metrics

Metric	Value
Sensitivity	0.9845
Specifity	0.3352
Balanced Accuracy	0.6598

Table 1: Performance Metrics for Logistic Regression

In Table 1, the performance measures of logistic regression obtained from the confusion matrix calculated on the test set are presented: the level of sensitivity appears to be high, indicating that our model can effectively distinguish safe water samples. However, the level of specificity is notably low, indicating that the model performs poorly in identifying cases where the water is unsafe.

3.2 Lasso

Our dataset encompasses a considerable number of variables, making the interpretation of logistic regression results quite challenging. Employing the LASSO (Least Absolute Shrinkage and Selection Operator) allows us to perform variable selection through a penalty term that tends to reduce the estimates of coefficients for less influential predictors towards zero. The LASSO minimizes the quantity of Equation (1) + the following quantity:

$$\lambda \sum_{j=1}^{p} |\beta_j| \tag{3}$$

The choice of the Lambda parameter is crucial as it impacts the coefficient estimates. A higher Lambda value leads to a greater reduction in variable coefficients. In our case, we selected the Lambda value using cross-validation to minimize the mean squared error (MSE). The Lambda value obtained from cross-validation is very small, at 0.00025, consequently retaining all variables in the dataset. The absence of reduction might indicate that all variables within the dataset play a significant role in predicting water safety.

3.3 Classification Trees

Classification trees enable the classification of observations by segmenting the predictor space. Trees are built using a top-down approach: the most discriminative predictor is selected, and its value is chosen to minimize the Gini index, which measures node purity and is represented in Equation 4.

$$\text{Gini Index} = \sum_{i=1}^{n} p_{mk} \cdot (1 - p_{mk}) \tag{4}$$

The process continues by selecting other variables, thus further dividing observations, until a stopping criterion is met. In the R package I used, the criteria include:

- A minimum of 20 observations per node;

- A minimum of 10 observations in a leaf (terminal node).

The resulting tree is depicted in Figure 5. Classification is based on the mode of the terminal node. For each node, at the top, there's a value that can be either 0 or 1, based on the modal class of the observations falling into that particular node, according to the "tests" indicated below the node. In the middle, the proportion of observations belonging to the modal class is represented, and at the bottom, the percentage of observations assigned to that specific node is depicted.
As an additional pruning criterion for the tree, the x-error (the mean of the misclassification error obtained through cross-validation) and the tree's complexity index were used. However, even based on this criterion, the ideal number of splits appears to be 13.

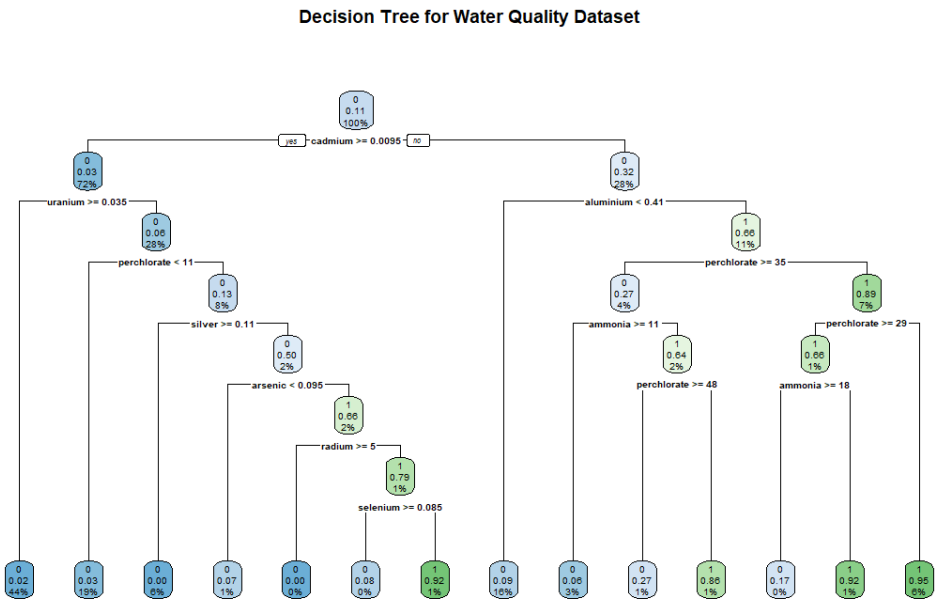

Figure 5: Classification Tree

3.3.1 Performance Metrics

Metric	Value
Sensitivity	0.9894
Specifity	0.6704
Balanced Accuracy	0.8299

Table 2: Performance Metrics for Classification Tree

The classification results showed Table 2 obtained through the tree are better compared to those achieved with logistic regression: once again, the model has no issues identifying cases where water is safe, and there has been a significant improvement in the specificity index, although it remains relatively low. Therefore, let's explore the final approach to ascertain if, with the available dataset, it is feasible to enhance the predictive quality of the model.

3.4 Random Forest

The last classification approach utilized was the random forest method. Essentially, instead of constructing a single tree on a training dataset, multiple estimations are obtained using B different training sets through the bagging technique. The average of these trees is computed to classify the observations. Additionally, for each tree, only a random subset of predictors (typically equal to the square root of the total predictors) is considered, ensuring that the same discriminative variables are not consistently selected.
Each predictor in a Random Forest model is assessed by its Mean Decrease Gini, which represents the average reduction in the Gini index when a specific predictor is used to split observations in the decision tree. This value indicates how much the predictor contributes to improving the purity of the tree's splits.
Thus, the analysis considered 50 trees arbitrarily, and the outcomes of the Random Forest are visualized in Figure 6.
Note that the most important variables are those also selected by the classification tree, although the reference order changes.

3.4.1 Performance Metrics

Metric	Value
Sensitivity	0.9901
Specifity	0.6983
Balanced Accuracy	0.8442

Table 3: Performance Metrics for Random Forest

This latest model represents a further improvement over the previous models in recognizing the 'unsafe' class. Unfortunately, however, the results cannot be considered satisfactory.

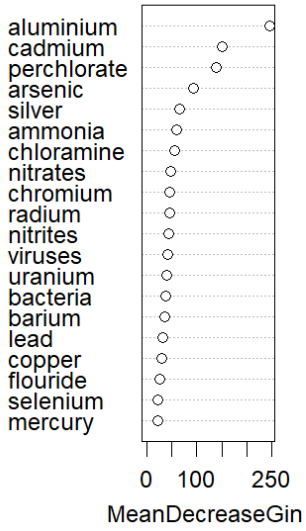

Figure 6: Mean Decrease Gini

3.5 Final remarks on Supervised Analysis

From our model results, it's evident that there's a lack of ability to identify cases where the water isn't safe. This could be due to the dataset being imbalanced. To address this issue, techniques like over-sampling or under-sampling could be utilized to rebalance the class distribution. Over-sampling involves generating additional observations for the underrepresented class, while under-sampling entails reducing observations from the overrepresented class. While these techniques might improve our results, caution is warranted as they could alter the reality or true distribution of the data (as class imbalances aren't uncommon). Given our goal was to understand model functioning, we opted against introducing these resampling techniques.

Regarding model interpretation, logistic regression and decision trees indeed offer better interpretability. Thus, predictive capability shouldn't be the sole parameter for selecting the best model; interpretability should be balanced with predictive accuracy when choosing the most suitable model.

4 Unsupervised Learning

Unsupervised learning aims to create a model capable of understanding the dataset's characteristics without including a response variable. For this type of analysis, we opted to employ the same dataset used in supervised approaches, excluding the 'is_safe' variable. Leveraging clustering models, the objective was to identify the set of features that differentiate various subsets of observations.

4.1 k-means Clustering

Unsupervised learning aims to create a model capable of understanding dataset characteristics that don't include a response variable. For this type of analysis, we chose to use the same dataset as in supervised learning, excluding the variable 'is_safe.' This allowed us to employ clustering models to find the set of features distinguishing different observation subgroups. The k-means clustering technique allows us to group observations into k clusters based on internal similarity within clusters and external dissimilarity between clusters. Distances between observations are measured using the squared Euclidean distance, which should be small within the same cluster and large between different clusters. The algorithm follows these steps:

1. Randomly assign observations to k clusters.
2. Calculate centroids for all k clusters, representing the average vector of p variables for observations in each cluster.
3. Assign each observation to the cluster whose centroid is closest.
4. Repeat steps 2 and 3 until convergence.

For curiosity, knowing that an initial distinction between different samples helps us understanding whether the water is safe or not, we tried using a reference number of clusters equal to 2. The composition of the different clusters will also depend on the random assignments in step 1. Therefore, the random assignment process will be repeated 20 times, choosing the clustering that provides the best Within-Cluster Sum of Squares for each cluster.
For k=2, we have one cluster with 3876 observations and another cluster with 4110 observations. The mean values of different variables (which have been standardized) for each cluster are presented in Table 4.
The ratio between between-cluster variance, i.e., the variation between clusters (which we want to maximize), and the total variance gives us a measure to understand how effective the clustering is. In our case, the variance explained by clustering over total variance is around 18%, which cannot be considered satisfactory.
The value of k was chosen, already aware of the existence of an underlying binary classification of the data. However, as this algorithm is often used without knowing the true nature of the data, we treated k as a hyperparameter. To determine k, we ran various k-means algorithms for k ranging from 1 to 10. The choice of k depends on the value of the total Within-Cluster Sum of Squares (WCSS): the problem is that this value decreases as the number of clusters increases. We need to find the k value where the WCSS decreases more gradually. To do this, we refer to a screeplot shown in Figure 7.

Variable	Cluster 1 Mean	Cluster 2 Mean
aluminium	0.5059	-0.4771
ammonia	0.1322	-0.1247
arsenic	0.4613	-0.4350
barium	0.6187	-0.5835
cadmium	-0.1987	0.1874
chloramine	0.8029	-0.7571
chromium	0.7453	-0.7028
copper	0.1425	-0.1343
flouride	0.0027	-0.0026
bacteria	0.2005	-0.1891
viruses	0.0054	-0.0051
lead	-0.0610	0.0575
nitrates	-0.0044	0.0040
nitrites	0.4820	-0.4545
mercury	-0.0015	0.0015
perchlorate	0.7705	-0.7266
radium	0.5043	-0.4756
selenium	0.0052	-0.0049
silver	0.7179	-0.6770
uranium	0.0057	-0.0054

Table 4: clusters' means for k=2

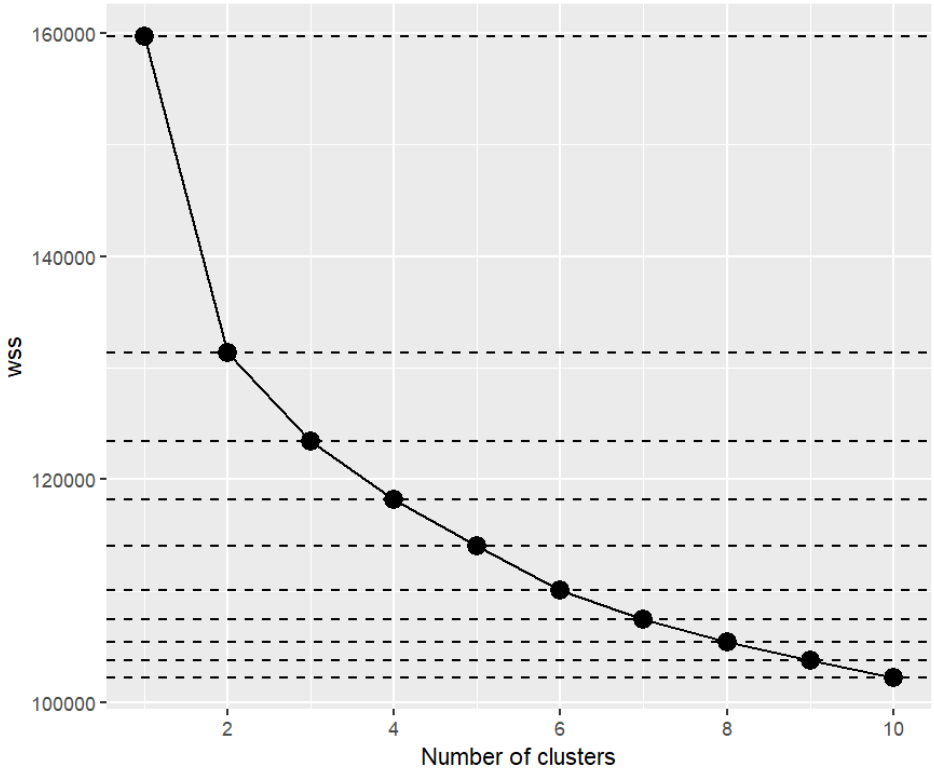

Figure 7: scree plot

We see that the ideal number of clusters could 2 or 3. Running the k-means algorithm for k=3, we obtain clusters with a respective number of observations of 1670, 2265, 4061, characterized by mean values for each variable indicated in Table 5.

Variable	**Cluster 1 Mean**	**Cluster 2 Mean**	**Cluster 3 Mean**
aluminium	0.4764	0.5159	-0.4825
ammonia	0.1426	0.1279	-0.1296
arsenic	1.6809	-0.4513	-0.4359
barium	0.8456	0.4317	-0.5865
cadmium	0.8140	-0.9566	0.2004
chloramine	0.8254	0.7721	-0.7680
chromium	0.7053	0.7643	-0.7146
copper	-0.1009	0.3219	-0.1382
flouride	0.0089	-0.0039	-0.0015
bacteria	0.0740	0.2918	-0.1930
viruses	0.0054	0.0008	-0.0027
lead	-0.2346	0.0678	0.0581
nitrates	0.0517	-0.0471	0.0051
nitrites	0.6856	0.3239	-0.4610
mercury	-0.0212	0.0113	0.0024
perchlorate	0.7672	0.7509	-0.7324
radium	0.4957	0.4989	-0.4809
selenium	0.0074	-0.0001	-0.0029
silver	0.6899	0.7168	-0.6818
uranium	0.0008	0.0077	-0.0046

Table 5: clusters' means for k=3

The variable 'arsenic' exhibits similar averages for clusters 2 and 3. Overall, there are few variables showing significantly different means across all clusters, specifically cadmium and nitrites. This could indicate overlapping clusters: indeed, the variance explained by the clustering for k=3 only slightly increases, reaching 22.7%
In our dataset, the k-means algorithm isn't yielding satisfactory results. Reflecting on the earlier univariate analysis, it's worth noting the presence of outliers in some variables. Given that k-means relies on means, these outliers might negatively impact the clustering outcomes. Hence, we might consider a reduced dataset by eliminating observations identified as outliers.

4.2 Hierarchical Clustering

Hierarchical clustering allows the identification of potential clusters through a dendrogram, a tree-like visualization. The dendrogram, structured in an ascending manner, displays observations at the base and progressively groups them based on their similarity. Mergers at the base of the tree involve more similar observations or groups of observations, while those at the end of the tree correspond to less similar ones. The distance, representing the similarity between clusters, can be calculated using different reference measures. In this case, we consider:

- The Average Linkage method, which calculates the average distance between clusters and merges based on the minimum of these average distances.
- The Complete Linkage method, which calculates the maximum distance between clusters and merges based on the minimum of these maximum distances.

The cutting of the dendrogram determines the number of clusters: a cut closer to the leaves corresponds to more clusters, while a cut farther from them implies fewer clusters.

Starting with the Average Linkage method, after standardizing the variables, the dendrogram's structure is obtained. Cutting the dendrogram at a height of 3, we observe a first cluster that includes 6822 observations, a second one of dimension 1161, and a third one including 3 observations. Considering a binary classification with around 7000 safe water samples and 900 unsafe ones, excluding the 15 observations from the third cluster suggests a similar distribution. However, it is necessary to compare the variable values to confirm the similarity. The variable means are listed in Table 6 which shows a significant diversity among them, unlike the k-means, where we chose k = 3 as hyperparameter.

Using the Complete Linkage method, the dendrogram reveals two clusters of observations. Cutting the dendrogram at a height of 2 yields two distinct clusters, with 4862 and 3124 observations respectively. The variable means for each cluster are presented in Table 7, indicating generally contrasting values between the two clusters.

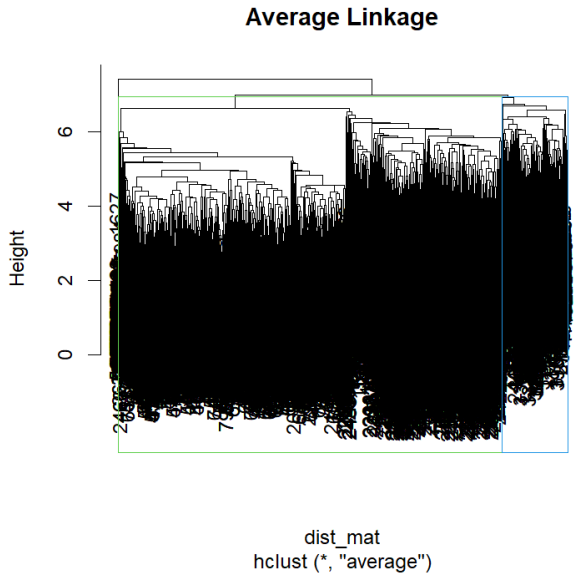

Figure 8: Average Linkage dendrogram

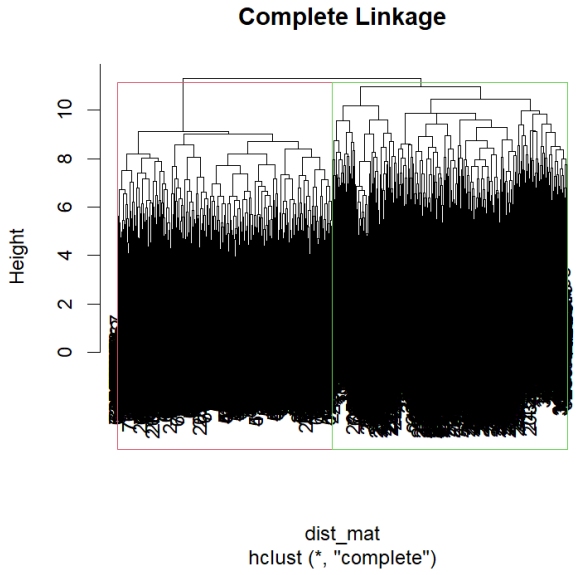

Figure 9: Complete Linkage dendrogram

Variable	Cluster 1 Mean	Cluster 2 Mean	Cluster 3 Mean
aluminium	-0.0863	3.2808	0.4984
ammonia	-0.0306	-0.1375	0.1801
arsenic	-0.3213	1.1457	1.8849
barium	-0.1370	1.1610	0.8022
cadmium	-0.1639	-0.6777	0.9646
chloramine	-0.1316	-0.6607	0.7747
chromium	-0.1146	0.5899	0.6718
copper	0.0223	-0.6367	-0.1297
flouride	0.0014	1.3583	-0.0115
bacteria	-0.0239	0.2646	0.1400
viruses	-0.0151	-0.1584	0.0892
lead	0.0423	0.2096	-0.2492
nitrates	-0.0053	-1.2514	0.0341
nitrites	-0.1164	0.8792	0.6817
mercury	0.0046	-1.4122	-0.0231
perchlorate	-0.1300	-0.3042	0.7647
radium	-0.0926	2.0279	0.5388
selenium	0.0062	0.0107	-0.0362
silver	-0.1198	-0.2391	0.7045
uranium	-0.0078	1.3134	0.0426

Table 6: clusters' means for k=3 with the average linkage method

Variable	Cluster 1 Mean	Cluster 2 Mean
aluminium	-0.3826	0.5955
ammonia	-0.0988	0.1537
arsenic	-0.4011	0.6243
barium	-0.4323	0.6728
cadmium	0.0393	-0.0612
chloramine	-0.5233	0.8144
chromium	-0.4849	0.7547
copper	-0.0676	0.1052
flouride	-0.0318	0.0495
bacteria	-0.0728	0.1133
viruses	0.0427	-0.0664
lead	0.0944	-0.1469
nitrates	0.0157	-0.0244
nitrites	-0.3232	0.5031
mercury	-0.0137	0.0213
perchlorate	-0.4784	0.7445
radium	-0.3416	0.5317
selenium	0e+00	1e-04
silver	-0.4912	0.7644
uranium	0.0058	-0.0091

Table 7: clusters' means for k=2 with the complete linkage method

4.3 Concluding Notes on Unsupervised Analysis

The k-means algorithm didn't perform well on our dataset, potentially due to certain variables displaying a long-tailed distribution, as observed during exploratory analysis. Given that k-means relies on computing Euclidean distances between observations, this could be significantly impacted by values in the tails of specific variables. One potential solution could involve transforming these variables, such as considering a logarithmic transformation.
This could mitigate the influence of extreme values and help improve the algorithm's performance by making the data more amenable to k-means clustering.
Alternatively, we could consider using a different distance measure for calculating distances between observations, such as the Manhattan distance or the Chebyshev distance. However, even in this scenario, it's crucial to thoroughly assess which measure is most suitable.
The creation of clusters in hierarchical clustering depends not only on the choice of distance measure but also on the criterion used to calculate the distance between clusters. This can lead to several advantages, offering greater flexibility and the potential to uncover more hidden patterns. However, it comes with the disadvantage that results depend on multiple parameters, which must be arbitrarily chosen. This arbitrary selection can make the clustering process less objective and more prone to subjective interpretations.