Xavier Institute of Engineering
Mahim, Mumbai 400016
Department of Information Technology
(Affiliated to University of Mumbai)

Class/ Sem / A.Y: BE IT/ VIII/ 2021-22

Course Name: R Programming Lab (ITL804)

Group No: 16

Water Potability Prediction

Student should be able to
ITL804.1: install and use R for simple programming tasks.
ITL804.2: estimate the functionality of R by using add-on packages.
ITL804.3: extract data from files and other sources and apply various data manipulation tasks on them.
ITL804.4: investigate statistical functions in R.
ITL804.5: summarize R Graphics to visualize results of various statistical operations on data.
ITL804.6: synthesize the knowledge of R gained to data Analytics for real life applications.

Rubrics For Laboratory Work

Roll No.	Name of the Student	Problem Statement (5)	Creativity & Quality of Work done (10)	Punctuality & lab ethics (5)	Performance/ Presentation (10)	Total (30)
28	Shrineeth Kotian					
29	Manish Kumavat					
40	Dixit Patel					

Ms. Jyotsna More
Subject Incharge

R Lab Mini Project
WATER POTABILITY PREDICTION

Shrineeth Kotian - XIEIT181925
Manish Kumavat - XIEIT181926
Dixit Patel -XIEIT181936

April 30, 2022

Contents

List of Figures

Water Quality Prediction:

1.1 Problem statement:

Water is an important aspect in our daily life which helps to regulate body temperature and maintain other bodily functions.Access to save drinking water is important for public health. According to the WHO can improved water supply and sanitation and better management of water resources boost the economic growth and contribute to poverty reduction. In addition, potable water is very important to maintain our bodily functions. A human body can survive up to 4 weeks without food, but only 3 days without water. Therefore, it is important to study which variables affects the potability of water.

1.2 Overview of Dataset:

The dataset in this study consists of 10 variables, with one dependent variable (1 = potable, 0 = not potable) and 9 independent variables. The independent variables are water parameters. The goal of this study is to predict potable water based on these water parameters.

Link to the dataset: `https://www.kaggle.com/datasets/adityakadiwal/water-potability`
Predicted Attribute: Potability
Number of Instances: 16380
Number Of Attributes: 10

1.2.0.1 Size of the dataset:

There are 16380 Rows and 11 Columns

```
Rows: 16,380
Columns: 11
$ ...1            <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, ~
$ ph              <dbl> NA, 3.716080, 8.099124, 8.316766, 9.092223, 5.584087, 10.223862, 8.635849, NA, 11.180284, 7.360640, 7.~
$ Hardness        <dbl> 204.8905, 129.4229, 224.2363, 214.3734, 181.1015, 188.3133, 248.0717, 203.3615, 118.9886, 227.2315, 16~
$ Solids          <dbl> 20791.32, 18630.06, 19909.54, 22018.42, 17978.99, 28748.69, 28749.72, 13672.09, 14285.58, 25484.51, 32~
$ Chloramines     <dbl> 7.300212, 6.635246, 9.275884, 8.059332, 6.546600, 7.544869, 7.513408, 4.563009, 7.804174, 9.077200, 7.~
$ Sulfate         <dbl> 368.5164, NA, NA, 356.8861, 310.1357, 326.6784, 393.6634, 303.3098, 268.6469, 404.0416, 326.6244, NA, ~
$ Conductivity    <dbl> 564.3087, 592.8854, 418.6062, 363.2665, 398.4108, 280.4679, 283.6516, 474.6076, 389.3756, 563.8855, 42~
$ Organic_carbon  <dbl> 10.379783, 15.180013, 16.868637, 18.436524, 11.558279, 8.399735, 13.789695, 12.363817, 12.706049, 17.9~
$ Trihalomethanes <dbl> 86.99097, 56.32908, 66.42009, 100.34167, 31.99799, 54.91786, 84.60356, 62.79831, 53.92885, 71.97660, 7~
$ Turbidity       <dbl> 2.963135, 4.500656, 3.055934, 4.628771, 4.075075, 2.559708, 2.672989, 4.401425, 3.595017, 4.370562, 3.~
$ Potability      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

Figure 1.1: Size and attributes of the Water Quality dataset

1.2.0.2 Attributes of our dataset:

1. Index,
2. pH value,
3. Hardness,
4. Solids (Total dissolved solids - TDS),
5. Chloramines,
6. Sulfate,
7. Conductivity,
8. Organic_carbon,
9. Trihalomethanes,
10. Turbidity,
11. Potability

1.2.0.3 Attributes details:

1. pH Value: PH is an important parameter in evaluating the acid-base balance of water. It also the indicator of acidic or alkaline condition of water status. WHO has recommended maximum permissible limit of pH from 6.5 to 8.5. The curent investigation ranges were 6.52-6.83 which are in the range of WHO standards.
2. Hardness: Hardness is mainly caused by calcium and magnesium salts. These salts are dissloved from geologic deposist thtough which water travels. The length of time water is in contact whit hardness producing meterial helps determine how muh hardness there is in raw water. Hardness was orginally defined as the capacity of water to prepitate soap caused by Calcium and Magnesium.
3. Solids(Total dissolved solids - TDS): Water has the ability to dissolve a wide range od inorganic and some organic minerals or salts such as potassium, calcium, sodium, bicarbonates, chlorides,magnesium, sulfates etc. These minerals produced un wanted taste and diluted color in appearence of water. This is the important paramater for the use of water. the water with high TDS value indicates that water is highly mineralized. Desirable limit for TDS is 500 mg/l and maximum limit is 1000mg/l which prescribed for drinking purpose.
4. Chloramines: Chlorine and chloramine are the major disnifetants used in public water systems. Chloloramines are most commonly formed when ammonia is added tho chlorine to treat drinking water. Chlorine levels up to 4 miligrams per liter (mg/L or 4 parts per million (ppm)) are considered safe in drinking water.
5. Sulfate: Sulfates are nanturaly occurring substances that are found in minerals, soil and roks. They are perents in ambient air, groundwater, plants and food. The principal commericial use of sulfate is in the chemical industry. Sulfate concentration in seawater is about 2.700 miligrams per liter(mg/l). It ranges from 3 to 20 mg/l in most freshwater suppiles, although much higher concentretions (100mg/l) are found in some geografic locations.
6. Conductivity: Pure water is not a good conductor of electiric current rether's a good insulator. Increase in ions concentration enahances the alectrical conductivity of water. Generaly, the amount of dissolved solids in water determines the electrical conductivity. Electrical conductivity (EC) actualy measures the iconic proces of a solution that enables it to transmit current. According to WHO standarts, EC calue should not exceeded 400 mikroS/cm.
7. Organic_carbon: Total Organic Carbon(TOC) in source waters comes from decaying natural

organic matter (NOM) as well as synthetic sources. TOC is a measure of the total amount of carbon in organic compounds in pure water. According to US EPA ¡ 2 mg/L as TOC in tread / drinking water, and ¡4mg/Lit int source water which is use for treatment.
8. Trihalomenthanes: THMs are chemicals which may be found in water treated with chlorine. The concentration of THMs in drinkin water varies according to the level of organic metarial in the water, the amount of chlorine required to treat the water, and the temperature od the water what is being treated. THM levels up to 80 ppm is considered safe in drinking water.
9. Turbidity: The turbidity od water depends on the quantity of solid matter present in the suspended state. It is a measure of light emitting properties of water and the test is used to indicate the quality od waste discharge with respect to colloided matter. The mean turbility value obtained for Wondo Genet Campus (0.98 NTU) is lower tha the WHO recomended value of 5.00 NTU.
10. Potability: Indicates is water is safe for human concumpition where 1 means Potable and 0 means Not potable

1.3 Purpose of the dataset:

- To determine the factors affecting the quality of water.
- To analyze the different parameters on which water potability depends.
- To find ways and methods to make water potable and safe for human consumption.

1.4 Steps in implementation of the project:

1. Perfrom Pre-Processing
2. Data Exploration (Statistical analysis of data and to find the relations between attributes)
3. Implementing Classification and Clustering Algorithms on our Water Quality Dataset.
4. Calculating accuracy, confusion matrix of all the algorithms.
5. Comparing the performance of all the Algorithms from each category and selecting the best Algorithm from each category for prediction of your selected dataset.

Implementation:

2.1 Loading the dataset:

We have performed our project on the RStudio platform, and the following figure will depict the loading of our dataset which is stored in the system.

```
library(tidyverse)
glimpse(water_potability2)
#READ DATA
WaterPotability <-
  read_csv("C:/Shiri c++/sem8/R_MINI/water_potability2.csv") %>%
  glimpse()
```

Figure 2.1: Loading the dataset

To implement further algorithms on the dataset, we convert the potability variable to factor variable.

```
Rows: 16,380
Columns: 11
$ x1              <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28,~
$ ph              <dbl> NA, 3.716080, 8.099124, 8.316766, 9.092223, 5.584087, 10.223862, 8.635849, NA, 11.180284, 7.360640, 7.974~
$ hardness        <dbl> 204.8905, 129.4229, 224.2363, 214.3734, 181.1015, 188.3133, 248.0717, 203.3615, 118.9886, 227.2315, 165.5~
$ solids          <dbl> 20791.32, 18630.06, 19909.54, 22018.42, 17978.99, 28748.69, 28749.72, 13672.09, 14285.58, 25484.51, 32452~
$ chloramines     <dbl> 7.300212, 6.635246, 9.275884, 8.059332, 6.546600, 7.544869, 7.513408, 4.563009, 7.804174, 9.077200, 7.550~
$ sulfate         <dbl> 368.5164, NA, NA, 356.8861, 310.1357, 326.6784, 393.6634, 303.3098, 268.6469, 404.0416, 326.6244, NA, 282~
$ conductivity    <dbl> 564.3087, 592.8854, 418.6062, 363.2665, 398.4108, 280.4679, 283.6516, 474.6076, 389.3756, 563.8855, 425.3~
$ organic_carbon  <dbl> 10.379783, 15.180013, 16.868637, 18.436524, 11.558279, 8.399735, 13.789695, 12.363817, 12.706049, 17.9278~
$ trihalomethanes <dbl> 86.99097, 56.32908, 66.42009, 100.34167, 31.99799, 54.91786, 84.60356, 62.79831, 53.92885, 71.97660, 78.7~
$ turbidity       <dbl> 2.963135, 4.500656, 3.055934, 4.628771, 4.075075, 2.559708, 2.672989, 4.401425, 3.595017, 4.370562, 3.662~
$ potability      <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
```

Figure 2.2: Coverting variable to factor variable

```
> summary(WaterPotability)
       x1              ph             hardness          solids         chloramines        sulfate        conductivity
 Min.   :    0   Min.   : 0.000   Min.   : 47.43   Min.   :  320.9   Min.   : 0.352   Min.   :129.0   Min.   :181.5
 1st Qu.: 4095   1st Qu.: 6.093   1st Qu.:176.85   1st Qu.:15666.7   1st Qu.: 6.127   1st Qu.:307.7   1st Qu.:365.7
 Median : 8190   Median : 7.037   Median :196.97   Median :20927.8   Median : 7.130   Median :333.1   Median :421.9
 Mean   : 8190   Mean   : 7.081   Mean   :196.37   Mean   :22014.1   Mean   : 7.122   Mean   :333.8   Mean   :426.2
 3rd Qu.:12284   3rd Qu.: 8.062   3rd Qu.:216.67   3rd Qu.:27332.8   3rd Qu.: 8.115   3rd Qu.:360.0   3rd Qu.:481.8
 Max.   :16379   Max.   :14.000   Max.   :323.12   Max.   :61227.2   Max.   :13.127   Max.   :481.0   Max.   :753.3
                 NA's   :2455                                                         NA's   :3905
 organic_carbon  trihalomethanes    turbidity      potability
 Min.   : 2.20   Min.   :  0.738   Min.   :1.450   0:9990
 1st Qu.:12.07   1st Qu.: 55.836   1st Qu.:3.440   1:6390
 Median :14.22   Median : 66.623   Median :3.955
 Mean   :14.28   Mean   : 66.396   Mean   :3.967
 3rd Qu.:16.56   3rd Qu.: 77.340   3rd Qu.:4.500
 Max.   :28.30   Max.   :124.000   Max.   :6.739
                 NA's   :810
```

Figure 2.3: Summary of Dataset

2.2 Pre-Processing:

Handling missing values is one of the common tasks in data analysis. In this, first we find the number of missing values in the dataset. And then comes the filling of the missing values using the methods such as mean, median, mode and other methods. We have used mean to fill the missing values.
In the dataset 3 columns have missing values. They are pH, sulfate, and trihalomethanes which have 2455, 3905, and 810 missing values respectively.

```
# A tibble: 1 x 11
     x1    ph hardness solids chloramines sulfate conductivity organic_carbon trihalomethanes turbidity potability
  <int> <int>    <int>  <int>       <int>   <int>        <int>          <int>           <int>     <int>      <int>
1     0  2455        0      0           0    3905            0              0             810         0          0
```

Figure 2.4: Missing Values

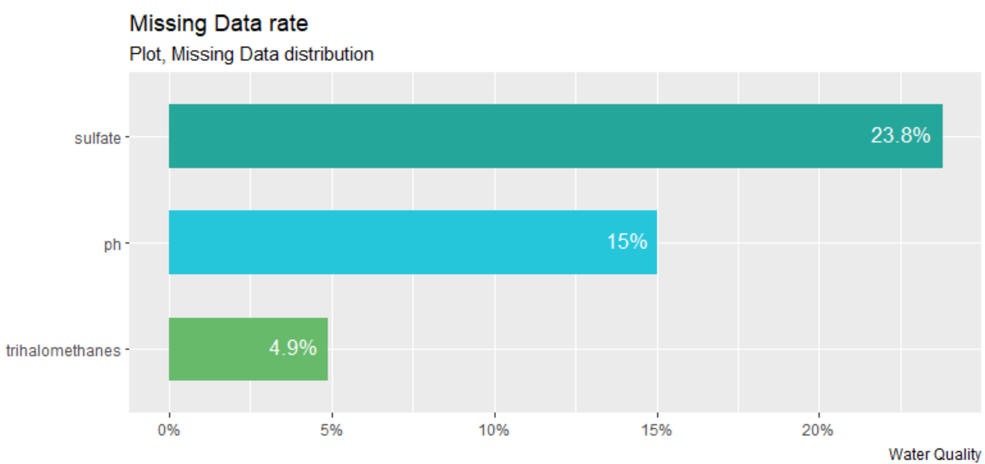

Figure 2.5: Missing Values graph

We plot a graph of missing values v/s target varibale i.e. potability

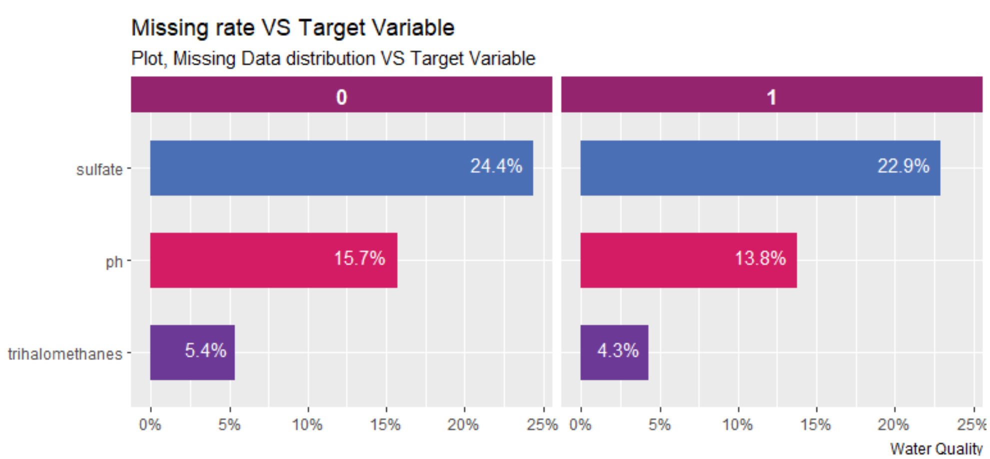

Figure 2.6: Missing values v/s Target varibale

We calculated mean value to handle mssing values of the dataset, and replced it wherever missing values were encountered.

```
# A tibble: 1 x 11
     x1    ph hardness solids chloramines sulfate conductivity organic_carbon trihalomethanes turbidity potability
  <int> <int>    <int>  <int>       <int>   <int>        <int>          <int>           <int>     <int>      <int>
1     0     0        0      0           0       0            0              0               0         0          0
```

Figure 2.7: Replaced Missing data with mean

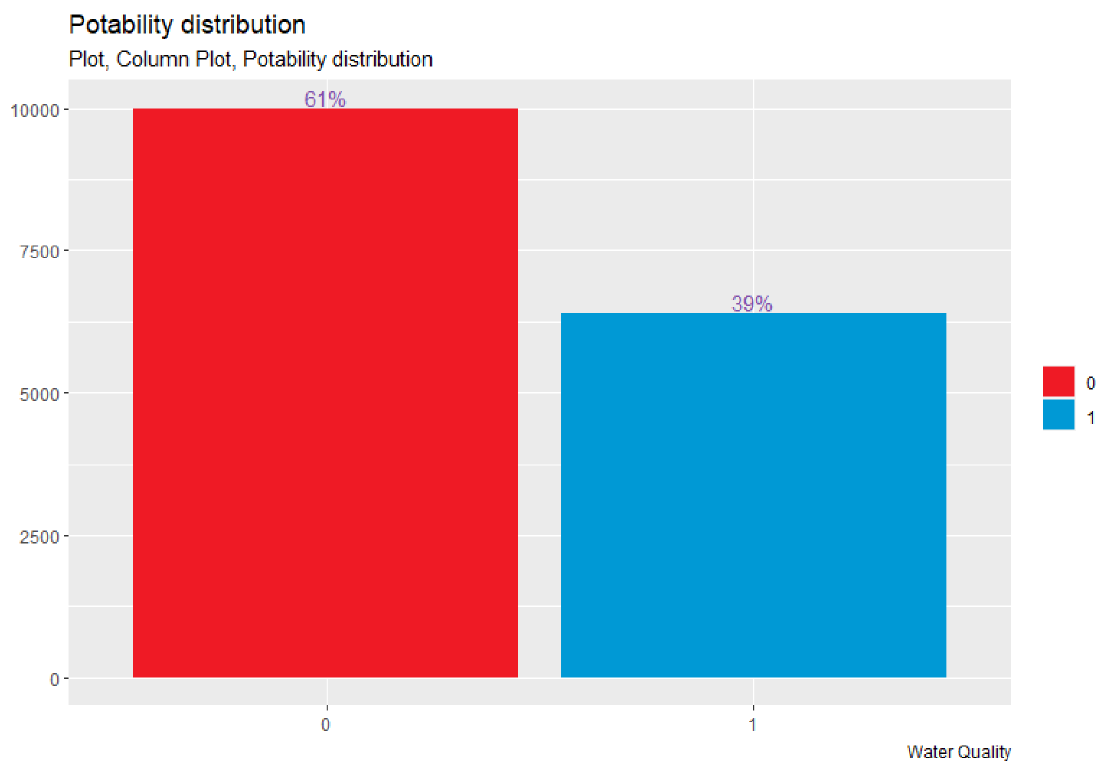

Figure 2.8: Potability variable after preprocessing

As we can see, our dataset is free from missing values which concludes the preprocessing part.

2.3 Exploratory Data Analysis:

Data exploration refers to the initial step in data analysis in which data analysts use data visualization and statistical techniques to describe dataset characterizations, such as size, quantity, and accuracy, in order to better understand the nature of the data. Data exploration can use a combination of manual methods and automated tools such as data visualizations, charts, and initial reports. We have used our full dataset for data exploration purpose.

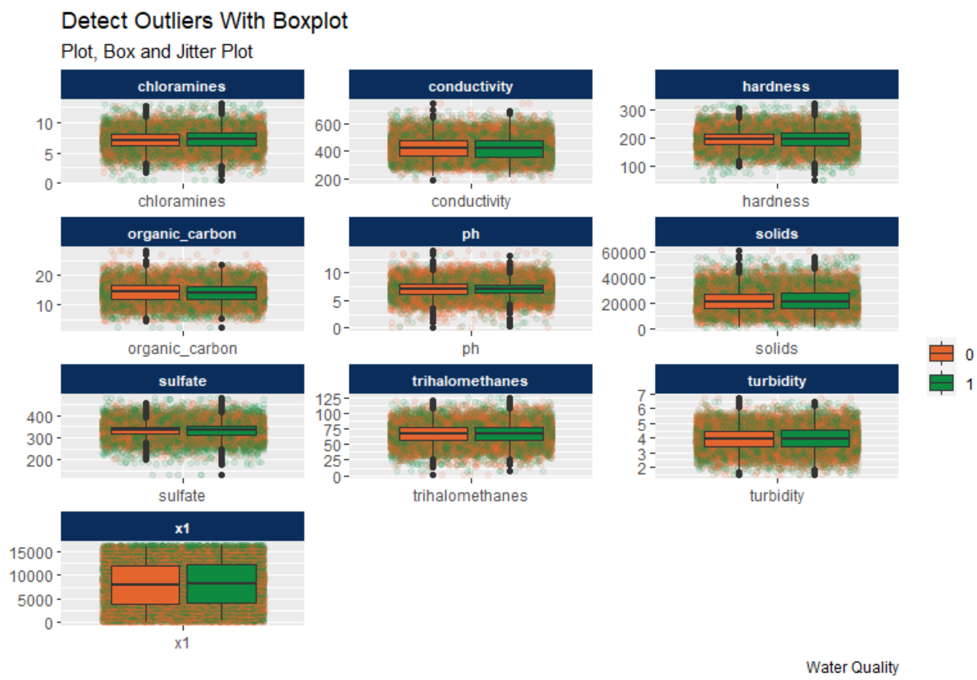

Figure 2.9: Detecting Outliers with Boxplot

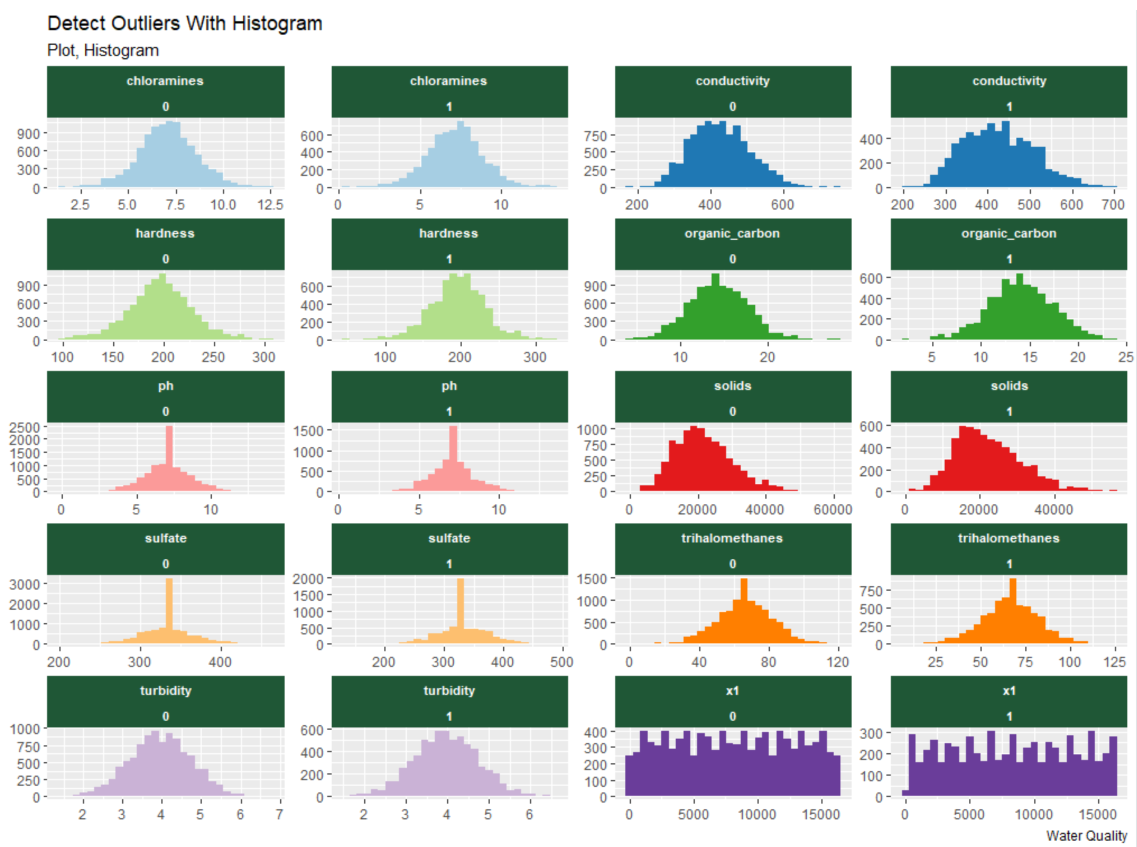

Figure 2.10: Detecting Outliers with Histogram

2.3.1 Correlation Matrix:

A correlation matrix is simply a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data.

A correlation matrix consists of rows and columns that show the variables. Each cell in a table contains the correlation coefficient.

In addition, the correlation matrix is frequently utilized in conjunction with other types of statistical analysis. For instance, it may be helpful in the analysis of multiple linear regression models. Remember that the models contain several independent variables. In multiple linear regression, the correlation matrix determines the correlation coefficients between the independent variables in a model.

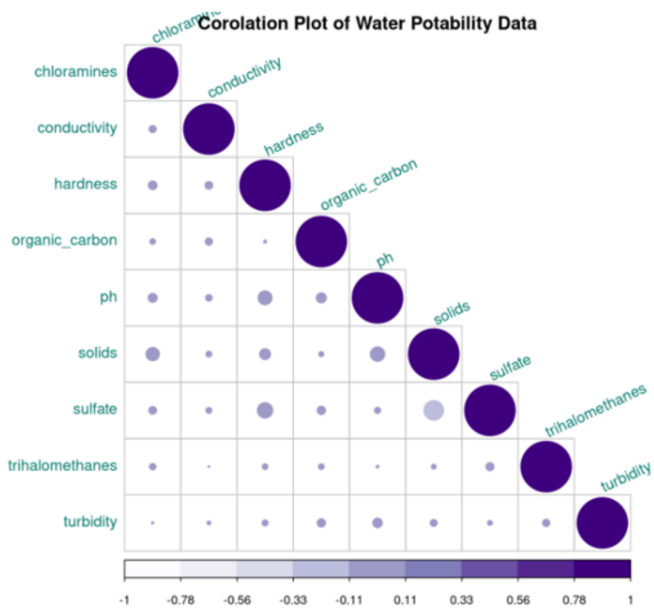

Figure 2.11: Correlation Matrix

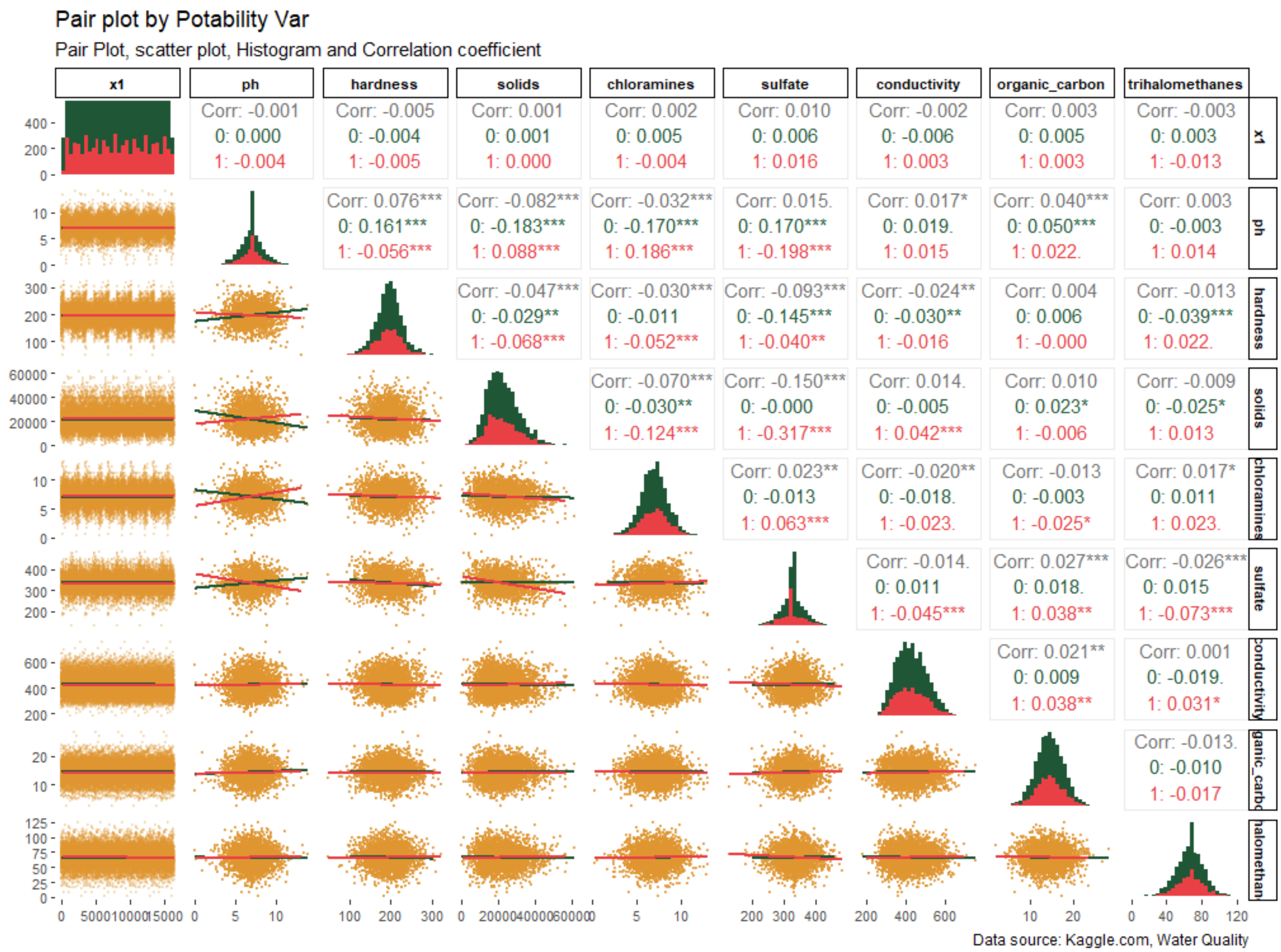

Figure 2.12: Pair Plot, Scatter Plot, Histogram, and Correlation Coefficient

A scatter plot matrix is a grid (or matrix) of scatter plots used to visualize bivariate relationships between combinations of variables. Each scatter plot in the matrix visualizes the relationship between a pair of variables, allowing many relationships to be explored in one chart.

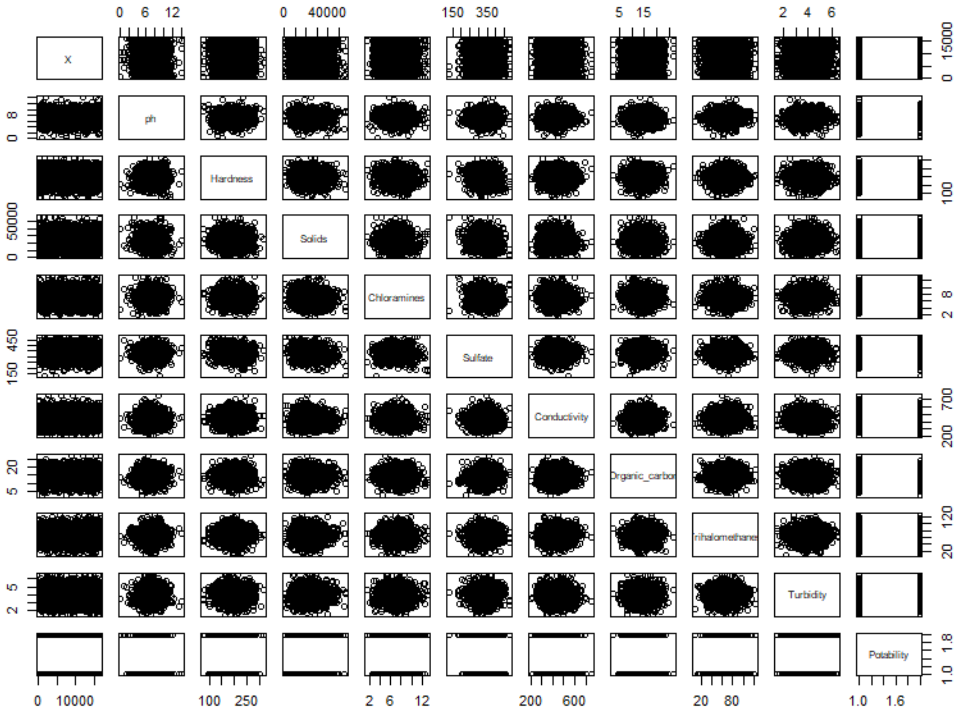

Figure 2.13: Scatterplot Matrix

The scatterplot matrix does not seem to show much association.

An Overlay Histogram allows you to visualize and compare multiple Populations superimposed on each other. In this Overlaid Histogram, we used melt() function to reshape the dataframe. The melt() function in R programming is an in-built function. It enables us to reshape and elongate the data frames in a user-defined manner. It organizes the data values in a long data frame format. Plotting the overall histograms reveals a bell-like shape for each of the variables in question with some possible skew in solids/conductivity.

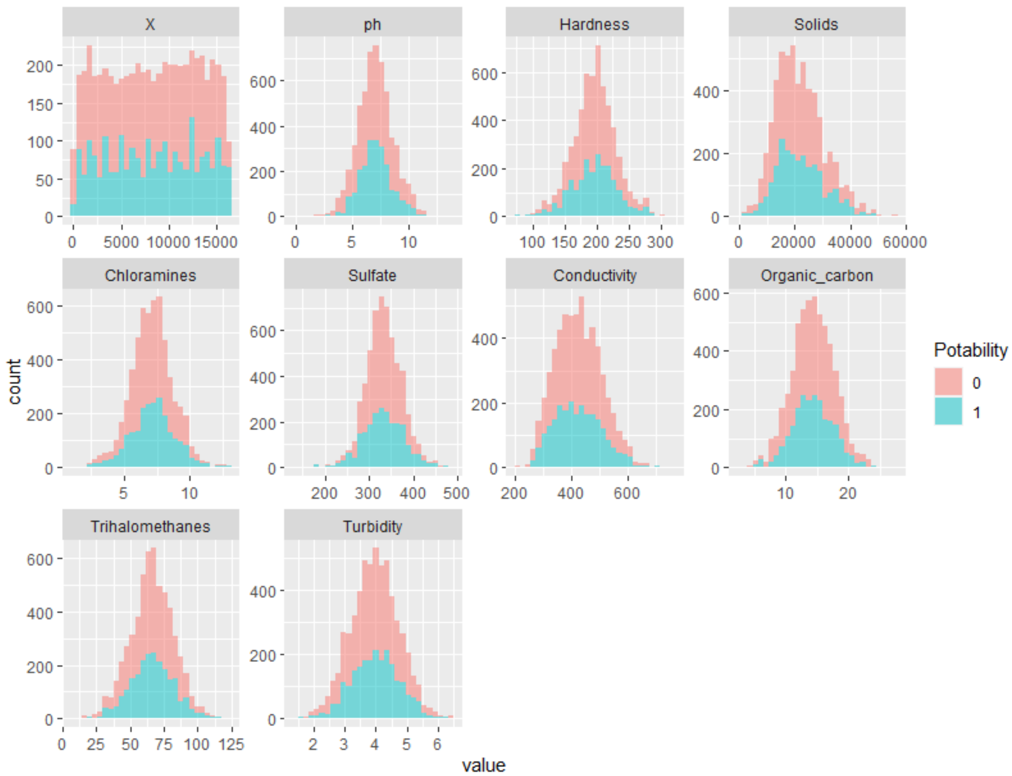

Figure 2.14: Overlaid Histograms

A violin plot is more informative than a plain box plot. While a box plot only shows summary statistics such as mean/median and interquartile ranges, the violin plot shows the full distribution of the data. The difference is particularly useful when the data distribution is multimodal (more than one peak).

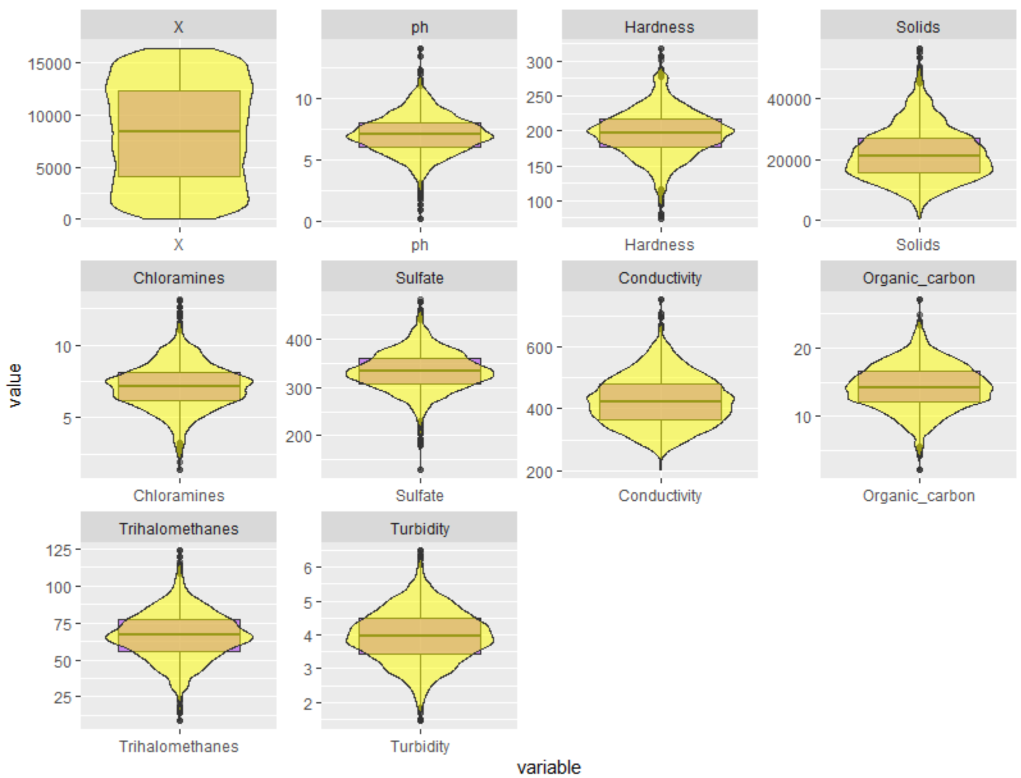

Figure 2.15: Violin and Boxplots

Again, the violin and boxplots show a symmetry in the distributions, but some skew in solids/ conductivity.
If solids/conductivity are related there shoiuld be some correlation between them, checking the correlation matrix quickly

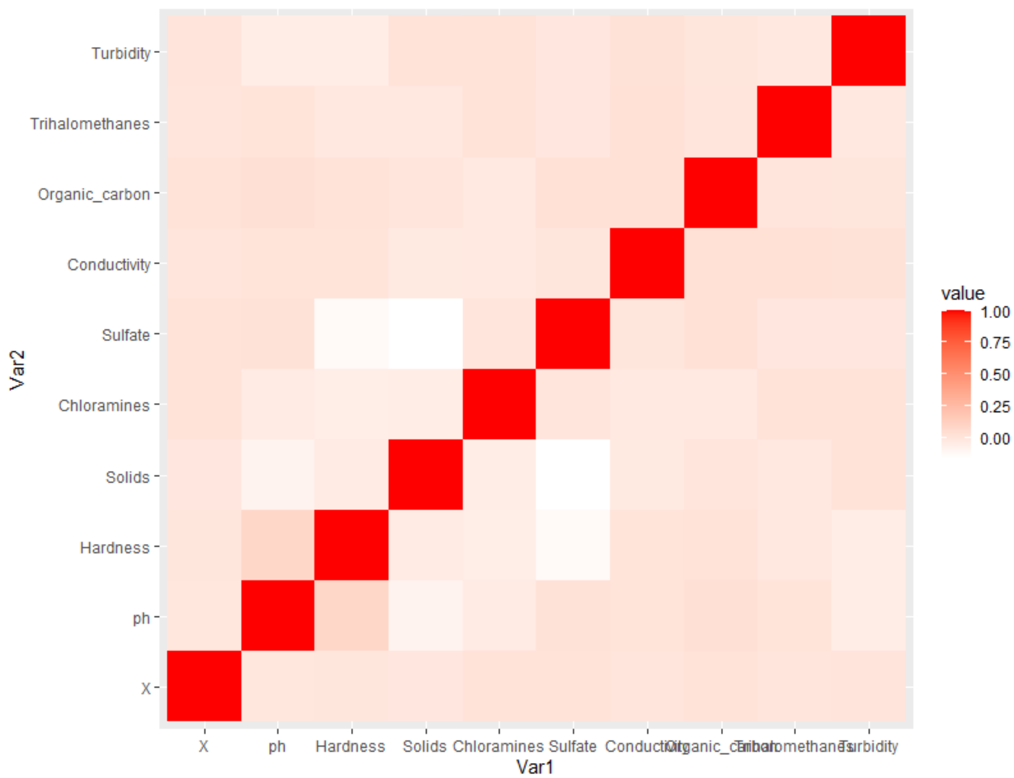

Figure 2.16: Correlation Matrix 2

This is a correlation matrix, which shows correlation among the variables.

A bar chart (aka bar graph, column chart) plots numeric values for levels of a categorical feature as bars. Levels are plotted on one chart axis, and values are plotted on the other axis. Each categorical value claims one bar, and the length of each bar corresponds to the bar's value. Bars are plotted on a common baseline to allow for easy comparison of values.

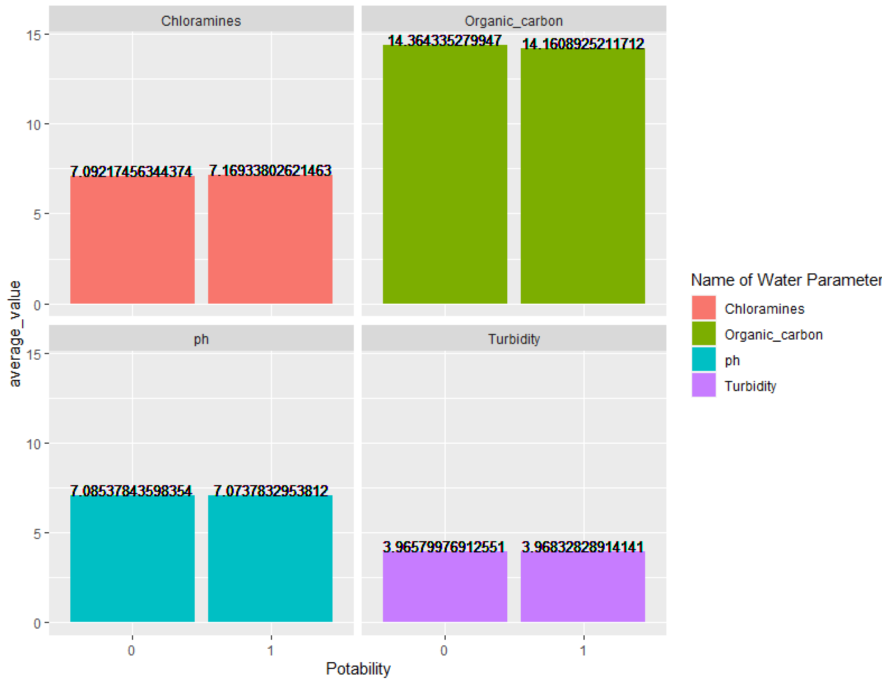

Figure 2.17: Bar Chart (Small Parameters)

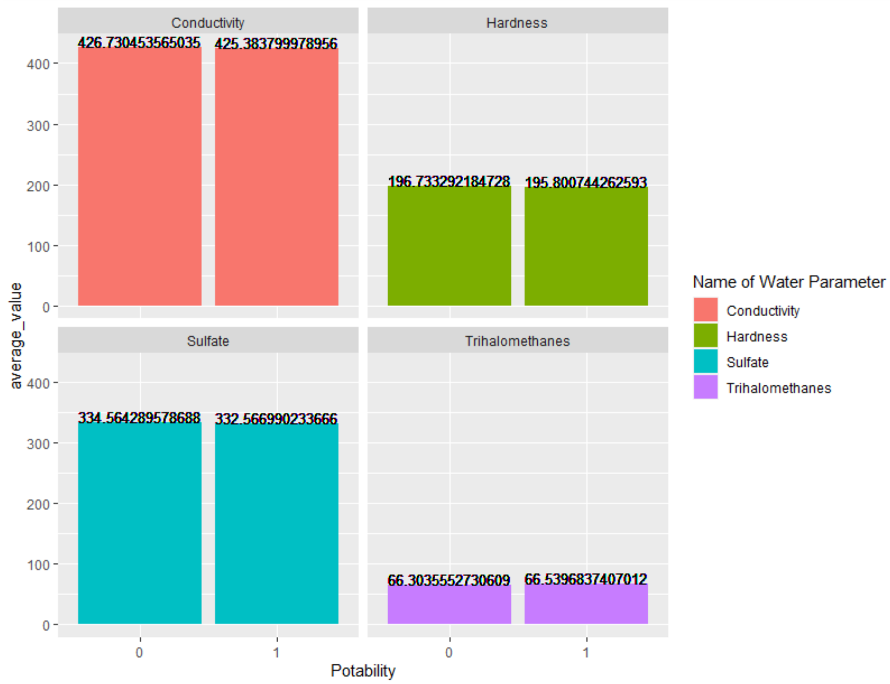

Figure 2.18: Bar Chart (Medium Parameters)

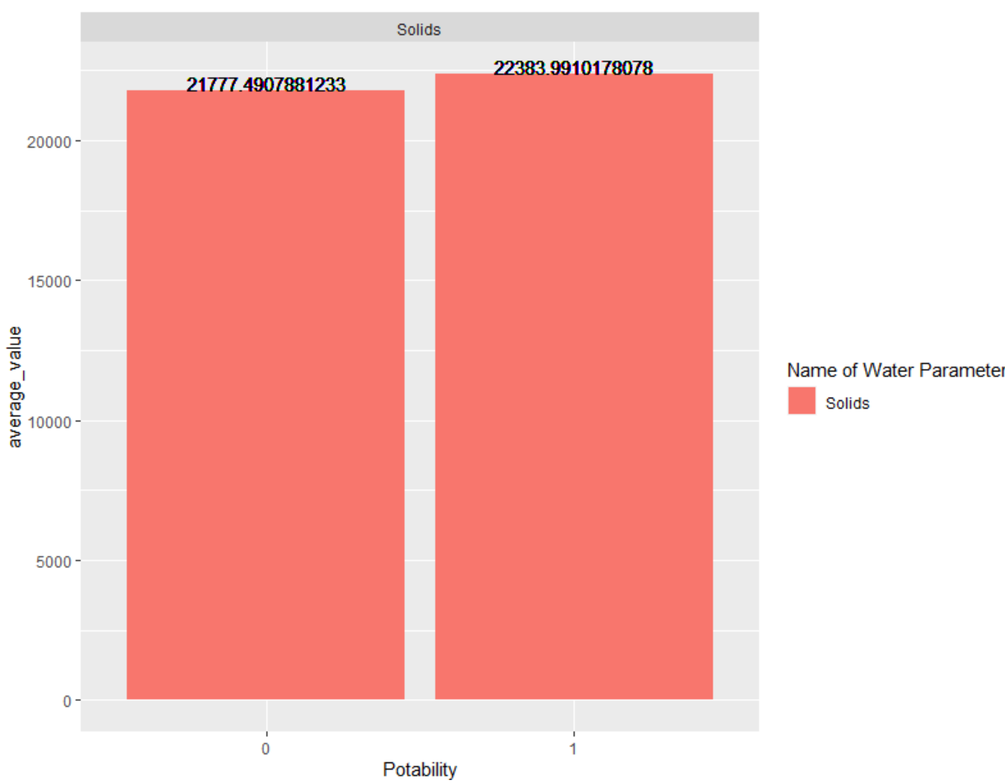

Figure 2.19: Bar Chart (Large Parameters)

2.4 Principal Component Analysis (PCA):

Principal Component Analysis (PCA) is a useful technique for exploratory data analysis, allowing you to better visualize the variation present in a dataset with many variables. It is particularly helpful in the case of "wide" datasets, where you have many variables for each sample. The most important use of PCA is to represent a multivariate data table as smaller set of variables (summary indices) in order to observe trends, jumps, clusters and outliers.

Figure 2.20: Scatterplot Graph

Importance of components:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Standard deviation	1.0949	1.0786	1.0229	1.0055	0.9983	0.9896	0.9823	0.93464	0.87513
Proportion of Variance	0.1332	0.1293	0.1163	0.1124	0.1107	0.1088	0.1072	0.09706	0.08509
Cumulative Proportion	0.1332	0.2625	0.3787	0.4911	0.6018	0.7106	0.8178	0.91491	1.00000

Figure 2.21: Principal Component Analysis

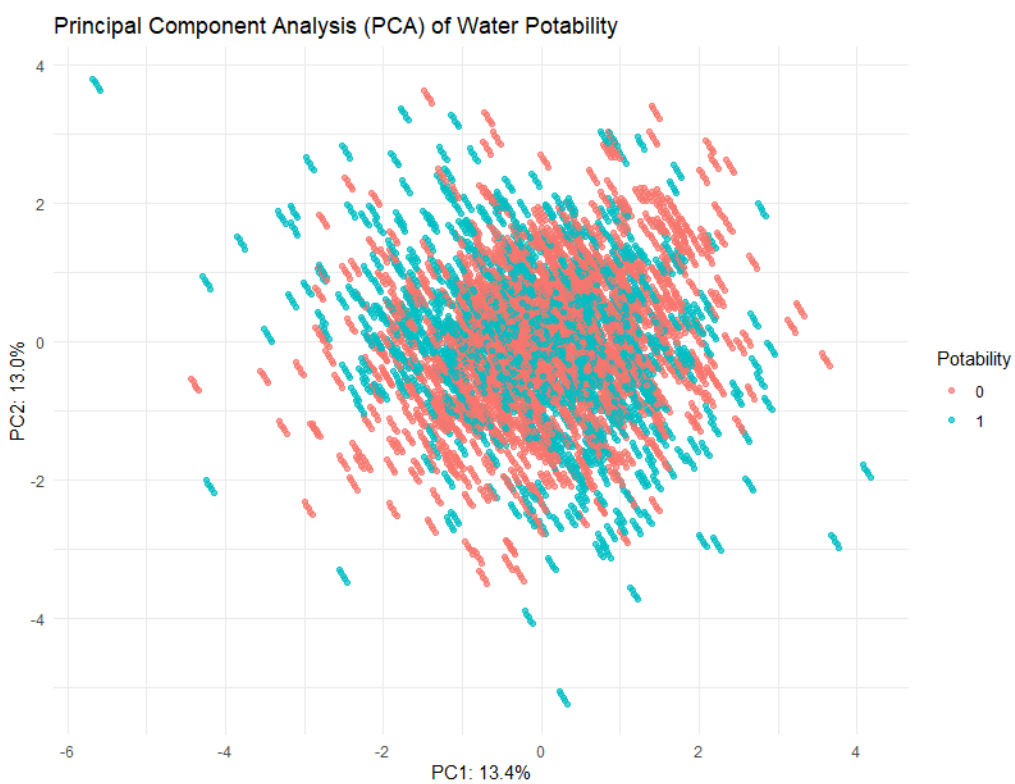

Figure 2.22: Principal Component Analysis using Scatterplot Graph

Box plots visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.

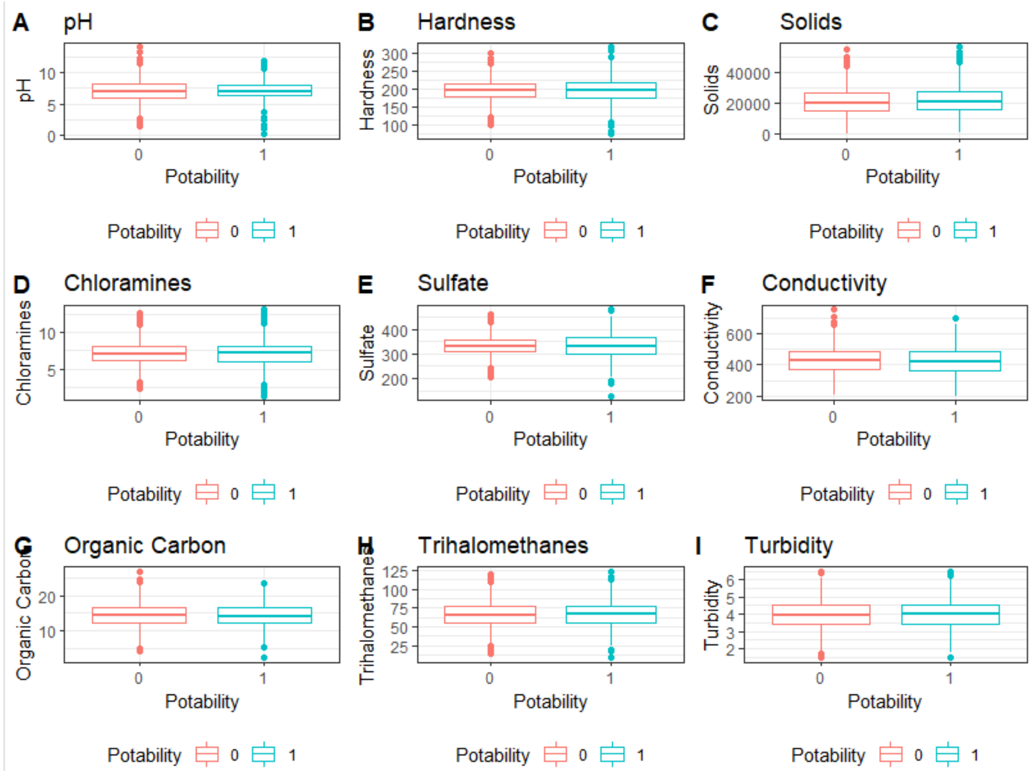

Figure 2.23: Boxplot Matrix of PCA values

2.5 DMBI/ML algorithms:

2.5.1 Classification Algorithms:

Classification means arranging the mass of data into different classes or groups on the basis of their similarities and resemblances. Classification plays an integral role in the context of mining techniques. As suggested by its name, this is a process where you classify data. And, many decisions need to be made to bring the data together. Often, it depends on a set of input variables. The classification depends on a series of acknowledgments and data instances.

2.5.1.1 Caret Random Forest Model:

We split our data in the dataset such that 80% is Training Data and 20% will be Testing Data. These are the summary of both the training and testing datas.

```
> summary(TrainingSet)
       x1              ph             hardness          solids          chloramines        sulfate
 Min.   :    0   Min.   : 0.000   Min.   : 47.43   Min.   :  320.9   Min.   : 0.352   Min.   :129.0
 1st Qu.: 4101   1st Qu.: 6.277   1st Qu.:176.90   1st Qu.:15671.1   1st Qu.: 6.124   1st Qu.:317.0
 Median : 8162   Median : 7.085   Median :197.06   Median :20944.6   Median : 7.132   Median :334.6
 Mean   : 8188   Mean   : 7.079   Mean   :196.37   Mean   :22025.0   Mean   : 7.124   Mean   :333.7
 3rd Qu.:12278   3rd Qu.: 7.865   3rd Qu.:216.67   3rd Qu.:27319.6   3rd Qu.: 8.124   3rd Qu.:350.4
 Max.   :16379   Max.   :14.000   Max.   :323.12   Max.   :61227.2   Max.   :13.127   Max.   :481.0
  conductivity   organic_carbon  trihalomethanes     turbidity      potability
 Min.   :181.5   Min.   : 2.20   Min.   :  0.738   Min.   :1.450   0:7992
 1st Qu.:365.5   1st Qu.:12.07   1st Qu.: 56.793   1st Qu.:3.437   1:5112
 Median :422.3   Median :14.19   Median : 66.304   Median :3.958
 Mean   :426.3   Mean   :14.27   Mean   : 66.458   Mean   :3.965
 3rd Qu.:481.8   3rd Qu.:16.52   3rd Qu.: 76.768   3rd Qu.:4.501
 Max.   :753.3   Max.   :28.30   Max.   :124.000   Max.   :6.739
```

Figure 2.24: Training Data

```
> summary(TestSet)
       x1              ph             hardness          solids          chloramines        sulfate
 Min.   :    8   Min.   : 0.000   Min.   : 73.49   Min.   :  728.8   Min.   : 0.352   Min.   :180.2
 1st Qu.: 4073   1st Qu.: 6.284   1st Qu.:176.68   1st Qu.:15629.8   1st Qu.: 6.149   1st Qu.:317.9
 Median : 8292   Median : 7.085   Median :196.69   Median :20866.5   Median : 7.118   Median :334.6
 Mean   : 8196   Mean   : 7.088   Mean   :196.36   Mean   :21970.6   Mean   : 7.117   Mean   :334.3
 3rd Qu.:12324   3rd Qu.: 7.895   3rd Qu.:216.66   3rd Qu.:27368.2   3rd Qu.: 8.074   3rd Qu.:350.4
 Max.   :16378   Max.   :14.000   Max.   :323.12   Max.   :61227.2   Max.   :13.127   Max.   :481.0
  conductivity   organic_carbon  trihalomethanes     turbidity      potability
 Min.   :181.5   Min.   : 2.20   Min.   :  8.176   Min.   :1.450   0:1998
 1st Qu.:368.3   1st Qu.:12.03   1st Qu.: 56.124   1st Qu.:3.463   1:1278
 Median :420.1   Median :14.31   Median : 66.304   Median :3.939
 Mean   :426.0   Mean   :14.33   Mean   : 66.145   Mean   :3.974
 3rd Qu.:481.9   3rd Qu.:16.67   3rd Qu.: 76.374   3rd Qu.:4.499
 Max.   :753.3   Max.   :27.01   Max.   :124.000   Max.   :6.739
```

Figure 2.25: Testing Data

Then we calculate mtry to check at which value will the model give less error

```
mtry = 3  OOB error = 0.04%
Searching left ...
mtry = 2        OOB error = 0.02%
0.4 1e-06
Searching right ...
mtry = 4        OOB error = 0.04%
-0.6666667 1e-06
```

Figure 2.26: Calculating MTRY

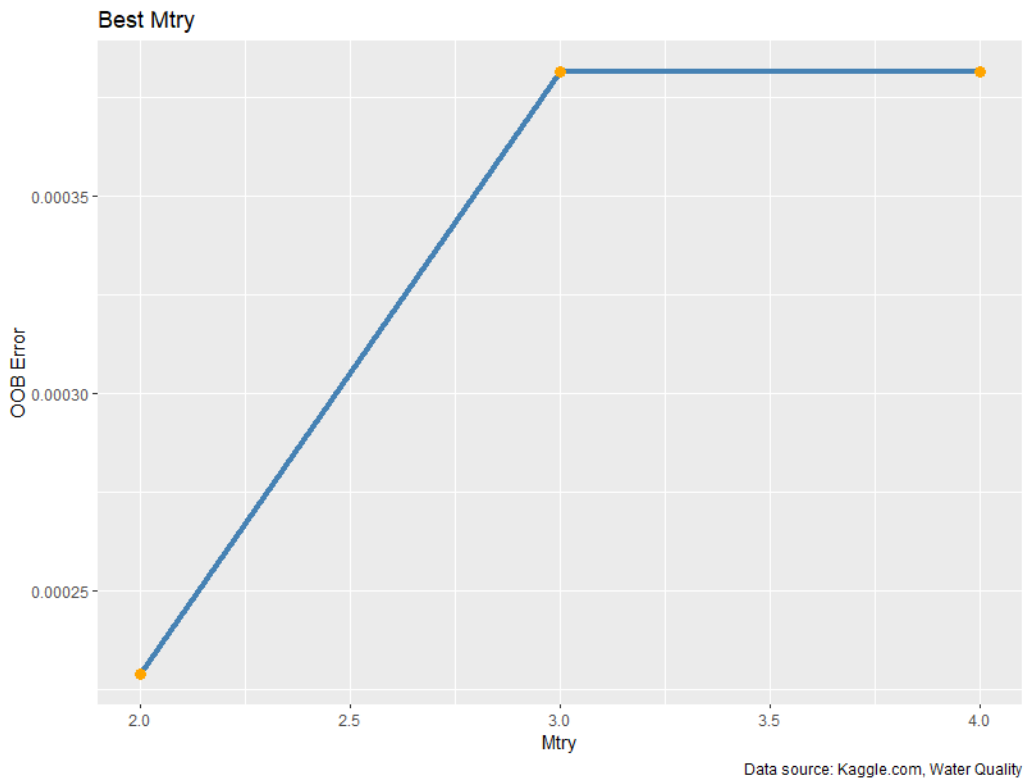

Figure 2.27: Displaying MTRY through line graph

```
Call:
 randomForest(formula = potability ~ ., data = TrainingSet, method = "rf",      metric = "Accuracy", tuneGrid = expand.grid(.mtry =
 4), trControl = control,      ntree = 1000)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 3

        OOB estimate of  error rate: 0.04%
Confusion matrix:
     0    1  class.error
0 7989    3 0.0003753754
1    2 5110 0.0003912363
```

Figure 2.28: Applying mtry to model

```
Call:
 randomForest(formula = potability ~ ., data = TrainingSet, ntree = 300,      mtry = 2, importance = TRUE, proximity = TRUE)
               Type of random forest: classification
                     Number of trees: 300
No. of variables tried at each split: 2

        OOB estimate of  error rate: 0.03%
Confusion matrix:
     0    1  class.error
0 7991    1 0.0001251251
1    3 5109 0.0005868545
```

Figure 2.29: Applying Best MTRY

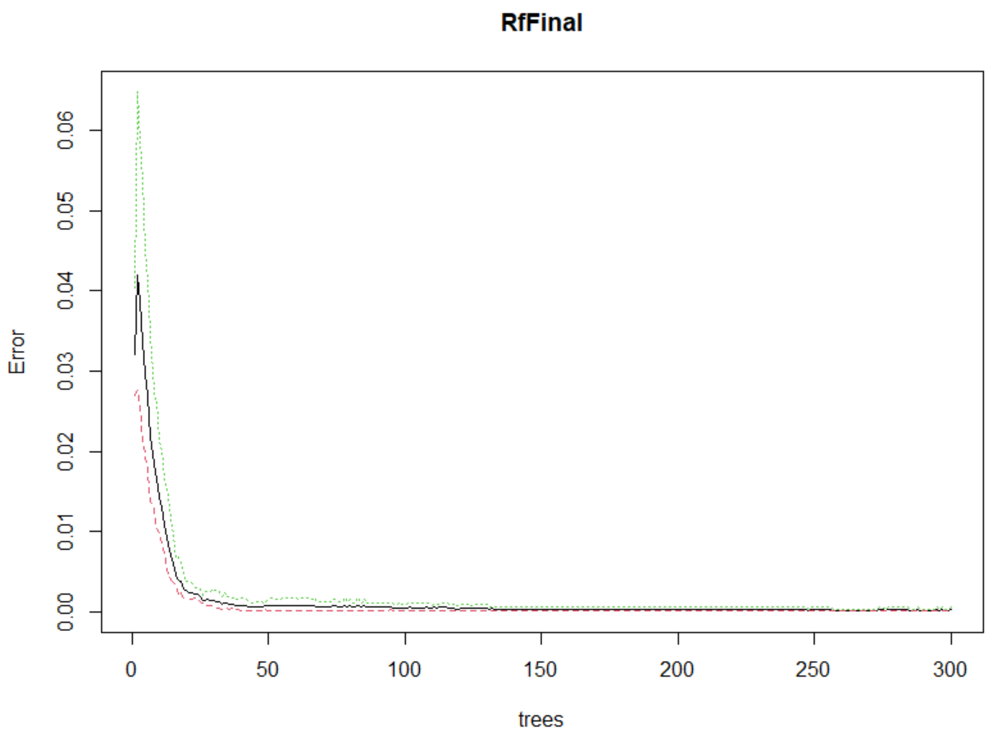

Figure 2.30: Visualizing Error Values

2.5.1.2 Logistic Regression:

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

```
Call:
glm(formula = Potability ~ ., family = "binomial", data = water_clean_train)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.204  -1.023  -0.959   1.328   1.564

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.295e+00  4.496e-01  -2.881  0.00397 **
X               1.125e-05  5.753e-06   1.956  0.05044 .
ph              3.484e-02  1.754e-02   1.986  0.04698 *
Hardness        3.389e-04  8.363e-04   0.405  0.68534
Solids          8.959e-06  3.211e-06   2.790  0.00527 **
Chloramines     3.698e-02  1.737e-02   2.129  0.03322 *
Sulfate        -1.903e-05  6.805e-04  -0.028  0.97769
Conductivity   -4.124e-04  3.368e-04  -1.225  0.22069
Organic_carbon -9.113e-03  8.233e-03  -1.107  0.26833
Trihalomethanes 1.033e-03  1.704e-03   0.606  0.54445
Turbidity       6.951e-02  3.517e-02   1.976  0.04812 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 7623.0  on 5654  degrees of freedom
Residual deviance: 7597.3  on 5644  degrees of freedom
AIC: 7619.3

Number of Fisher Scoring iterations: 4
```

Figure 2.31: Building Logistic Regression Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1112  762
         1    1   11

               Accuracy : 0.5954
                 95% CI : (0.5729, 0.6177)
    No Information Rate : 0.5901
    P-Value [Acc > NIR] : 0.3286

                  Kappa : 0.0157

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.014230
            Specificity : 0.999102
         Pos Pred Value : 0.916667
         Neg Pred Value : 0.593383
             Prevalence : 0.409862
         Detection Rate : 0.005832
   Detection Prevalence : 0.006363
      Balanced Accuracy : 0.506666

       'Positive' Class : 1
```

Figure 2.32: Tuning Performance of Logistic Regression Model

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1506  998
         1    2    8

               Accuracy : 0.6022
                 95% CI : (0.5828, 0.6214)
    No Information Rate : 0.5998
    P-Value [Acc > NIR] : 0.4119

                  Kappa : 0.0079

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.007952
            Specificity : 0.998674
         Pos Pred Value : 0.800000
         Neg Pred Value : 0.601438
             Prevalence : 0.400159
         Detection Rate : 0.003182
   Detection Prevalence : 0.003978
      Balanced Accuracy : 0.503313

       'Positive' Class : 1
```

Figure 2.33: Testing Performance of Logistic Regression Model

2.5.1.3 Decision tree:

The Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1102  705
         1   11   68

               Accuracy : 0.6204
                 95% CI : (0.598, 0.6423)
    No Information Rate : 0.5901
    P-Value [Acc > NIR] : 0.00397

                  Kappa : 0.0905

 Mcnemar's Test P-Value : < 2e-16

            Sensitivity : 0.08797
            Specificity : 0.99012
         Pos Pred Value : 0.86076
         Neg Pred Value : 0.60985
             Prevalence : 0.40986
         Detection Rate : 0.03606
   Detection Prevalence : 0.04189
      Balanced Accuracy : 0.53904

       'Positive' Class : 1
```

Figure 2.34: Decision Tree Classifier model (Tuning Performance)

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1500  911
         1    8   95

               Accuracy : 0.6344
                 95% CI : (0.6153, 0.6533)
    No Information Rate : 0.5998
    P-Value [Acc > NIR] : 0.0002014

                  Kappa : 0.1048

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.09443
            Specificity : 0.99469
         Pos Pred Value : 0.92233
         Neg Pred Value : 0.62215
             Prevalence : 0.40016
         Detection Rate : 0.03779
   Detection Prevalence : 0.04097
      Balanced Accuracy : 0.54456

       'Positive' Class : 1
```

Figure 2.35: Decision Tree Classifier model (Testing Performance)

2.5.1.4 Random Forest Model:

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1101    9
         1   12  764

               Accuracy : 0.9889
                 95% CI : (0.983, 0.9931)
    No Information Rate : 0.5901
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.977

 Mcnemar's Test P-Value : 0.6625

            Sensitivity : 0.9884
            Specificity : 0.9892
         Pos Pred Value : 0.9845
         Neg Pred Value : 0.9919
             Prevalence : 0.4099
         Detection Rate : 0.4051
   Detection Prevalence : 0.4115
      Balanced Accuracy : 0.9888

       'Positive' Class : 1
```

Figure 2.36: Random Forest Model (Tuning Performance)

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1494   11
         1   14  995

               Accuracy : 0.9901
                 95% CI : (0.9854, 0.9936)
    No Information Rate : 0.5998
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.9793

 Mcnemar's Test P-Value : 0.6892

            Sensitivity : 0.9891
            Specificity : 0.9907
         Pos Pred Value : 0.9861
         Neg Pred Value : 0.9927
             Prevalence : 0.4002
         Detection Rate : 0.3958
   Detection Prevalence : 0.4014
      Balanced Accuracy : 0.9899

       'Positive' Class : 1
```

Figure 2.37: Random Forest Model (Testing Performance)

This Random Forest Model gives the highest accuracy untill now of 99.01% .

2.5.2 Visualizations in R:

We have made 4 models of algorithms such as Random Forest, XGBOOST, K-Nearest Neighbor(KNN) and Logistic Regression models and compared its accuracy and specificity that predicts potabilty.

KNN is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it are in.

This KNN model is perhaps the most adapted for our situation : indeed, the aim of this process is to determine the K-closest observations of each input (given K the number of observations to determine, and some metrics that define the distance evaluation to find the "neighbors"). Then the model will simply decide to classify the input in the class that most appears amongs its "neighbors" class.

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data (images, text, etc.) artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree based algorithms are considered best-in-class right now.

```
# A tibble: 6 x 11
# Groups:   Potability [1]
      X    ph Hardness Solids Chloramines Sulfate
  <int> <dbl>    <dbl>  <dbl>       <dbl>   <dbl>
1     0  7.09     205. 20791.        7.30    369.
2     1  3.72     129. 18630.        6.64    335.
3     2  8.10     224. 19910.        9.28    335.
4     3  8.32     214. 22018.        8.06    357.
5     4  9.09     181. 17979.        6.55    310.
6     5  5.58     188. 28749.        7.54    327.
# ... with 5 more variables: Conductivity <dbl>,
#   Organic_carbon <dbl>, Trihalomethanes <dbl>,
#   Turbidity <dbl>, Potability <fct>
```

Figure 2.38: Converting our target variable to categorical data

```
# A tibble: 2 x 2
  Potability     n
  <fct>      <int>
1 0           9990
2 1           6390
```

Figure 2.39: Summary of Potability variable

We split the dataset into 80% training data and 20%testing data and will make the models out of it.

```
Call:
 randomForest(formula = Potability ~ ., data = trn_water, ntree = 1000)
               Type of random forest: classification
                     Number of trees: 1000
No. of variables tried at each split: 3

        OOB estimate of  error rate: 0.04%
Confusion matrix:
     0    1  class.error
0 7989    3 0.0003753754
1    2 5110 0.0003912363
```

Figure 2.40: Building RF Model

```
k-Nearest Neighbors

13104 samples
   10 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 10484, 10484, 10484, 10482, 10482
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.5454824  -0.004542125
  7  0.5282351  -0.054314457
  9  0.5240377  -0.073618933

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

Figure 2.41: Building KNN Model

```
eXtreme Gradient Boosting

13104 samples
   10 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 11794, 11793, 11794, 11793, 11793, 11794, ...
Resampling results across tuning parameters:

  eta  max_depth  colsample_bytree  subsample  nrounds  Accuracy   Kappa
  0.3  1          0.6               0.50        50      0.7389340  0.3922984
  0.3  1          0.6               0.50       100      0.7631242  0.4568191
  0.3  1          0.6               0.50       150      0.7705265  0.4784586
  0.3  1          0.6               0.75        50      0.7320653  0.3719529
  0.3  1          0.6               0.75       100      0.7603014  0.4471657
  0.3  1          0.6               0.75       150      0.7714422  0.4785279
  0.3  1          0.6               1.00        50      0.7301574  0.3632343
```

Figure 2.42: Building XGBOOST Model

```
Generalized Linear Model

13104 samples
   10 predictor
    2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 10484, 10484, 10484, 10482, 10482
Resampling results:

  Accuracy   Kappa
  0.6106535  0.002812279
```

Figure 2.43: Building Logistic Regression Model

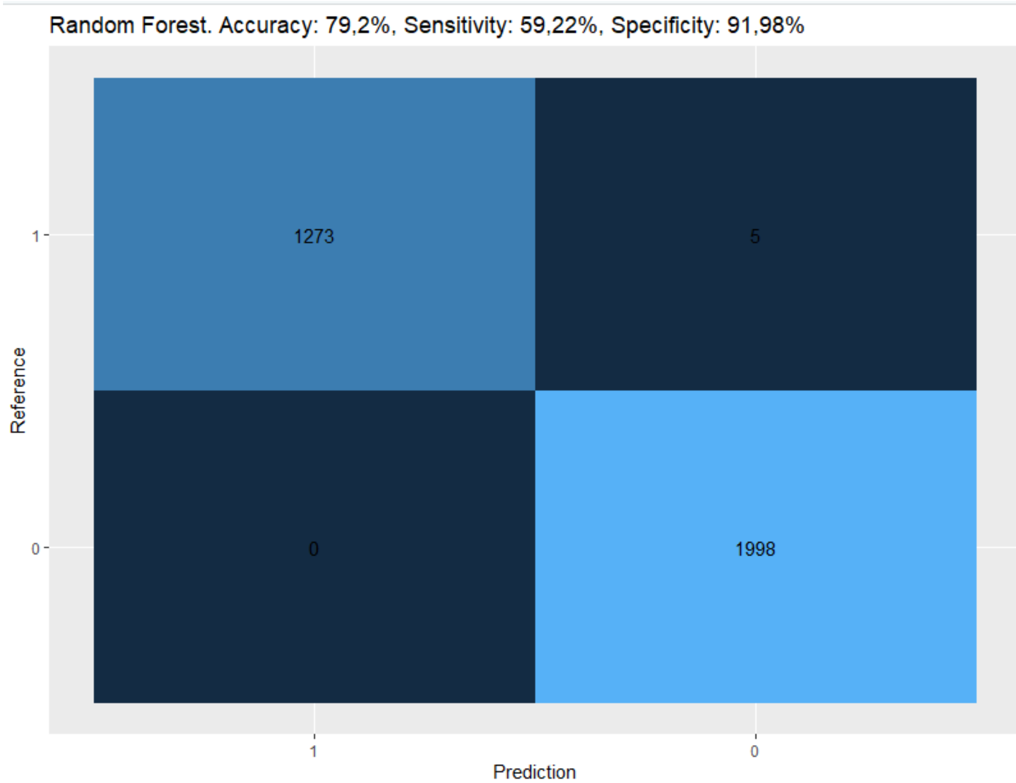

Figure 2.44: Confusion Matrix Plot for Random Forest Model

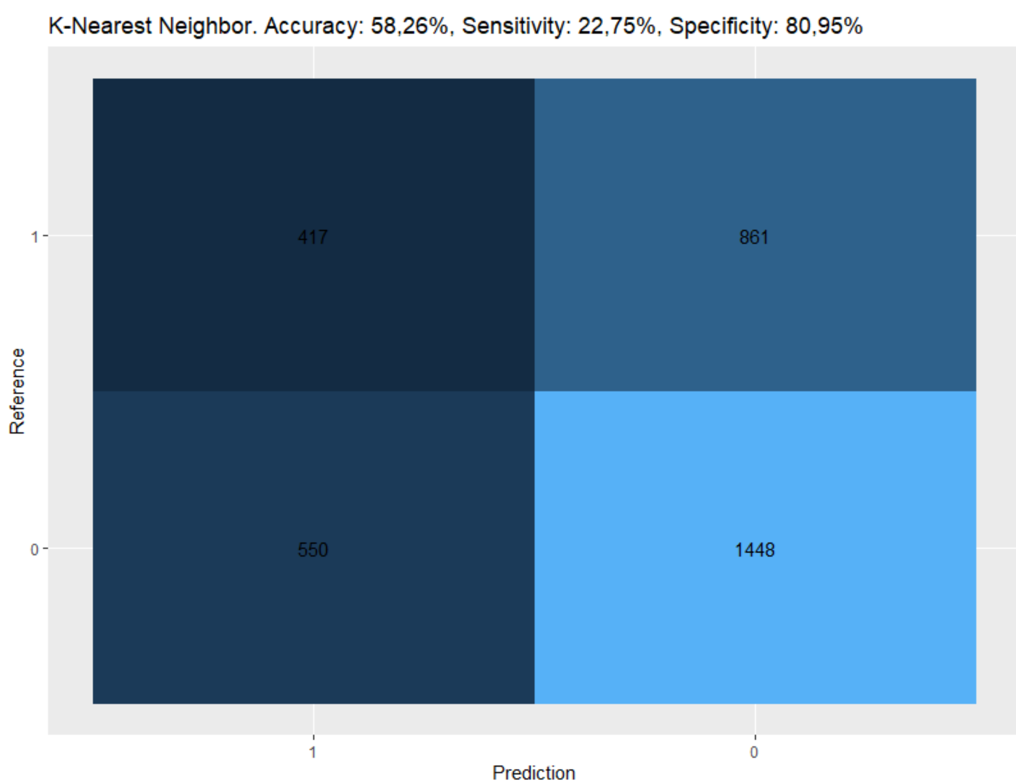

Figure 2.45: Confusion Matrix Plot for KNN Model

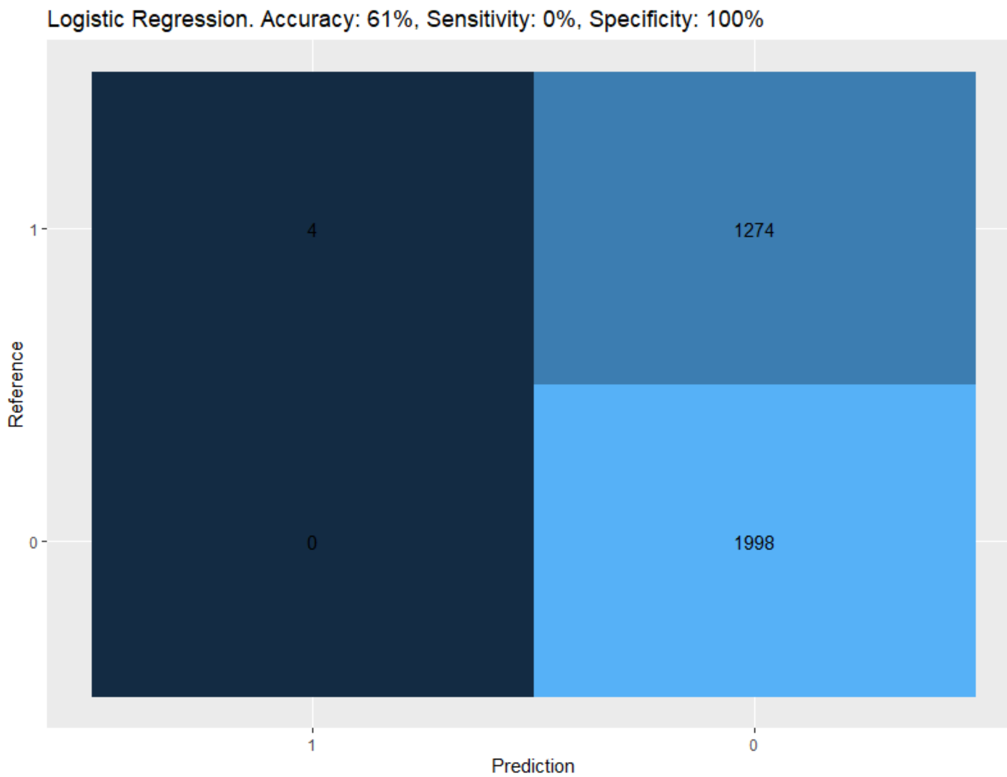

Figure 2.46: Confusion Matrix Plot for Logistic Regression Model

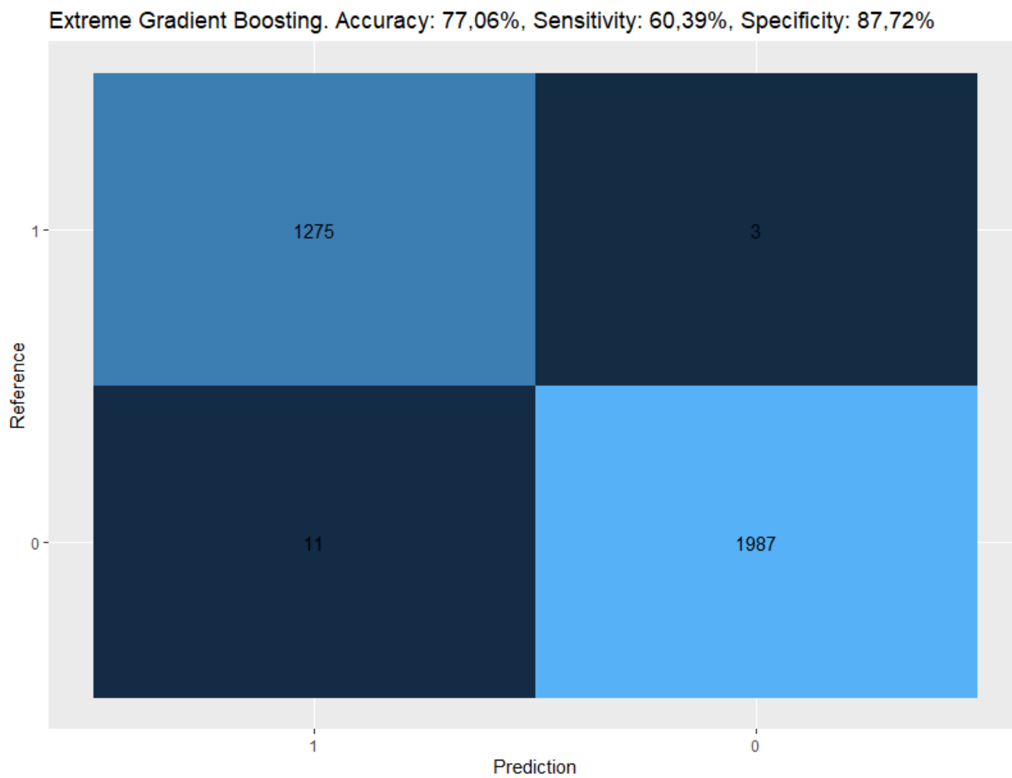

Figure 2.47: Confusion Matrix Plot for XGBOOST Model

Conclusion

The logistic regression model seems to fail to find a pattern in the data. It predicts that everything is not drinkable. A lower threshold (i.e. lower than 0.5) may improve this model. The best performing model is the random forest: it has a test accuracy of 79,2 %. Predicting not drinkable water as drinkable (false positive) is in my opinion the most crucial thing to avoid. Therefore, specificity is the most important measure. Because, a high specificity means many true negatives and few false positives. Random forest outperforms the other models with a specificity of 91,98 %. The Highest accuracy we calculated was of Random Forest Model that is 99.01%. The accuracies obtained are quite encouraging and are as follows:

- 60.22 % for the Logistic Regression Model.
- 63.44 % for the Decision Tree Classifier
- 99.01 % for the Random Forest Model.
- 58.26 % for the K-Nearest Neighbors Classifier
- 77.06 % for the Extreme Gradient Boosting

Hence we could predict the Potability of water with the help of the factors provided with the highest accuracy of 99.01%.

References

1. https://www.kaggle.com/tristan581/17k-apple-app-store-strategy-games
2. https://www.youtube.com/watch?v=eq1zKgCFwkk&t=306s
3. https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html#:~:text=low%20point%20density.-,Density%2DBased%20Spatial%20Clustering%20of%20Applications%20with%20Noise%20(DBSCAN),is%20containing%20noise%20and%20outliers.
4. https://www.javatpoint.com/data-mining-techniques
5. https://www.geeksforgeeks.org/basic-concept-classification-data-mining/#:~:text=In%20the%20process%20of%20data,distinguishes%20data%20classes%20and%20concepts.