Water Potability (CYO Project (HarvardX))

Ghodsieh Ghanbari

7/27/2021

Contents

Overview

Current project is related to the Choose Your Own project of the Capstone course of Harvardx data science professional certificate. I decided to use water quality data set for this project. This data set contains several attributes of water including PH, hardness, solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, and Turbidity which can determine if water is potable or not. Based on the data, we are dealing with a binary classification data. Therefore, the aim of this project is to perform some machine learning models to predict whether water is drinkable or not by having some characteristics of water.

Reading the data set and applying required packages

```
maindata <-read.csv("water_potability.csv")
```

```
## Loading required package: tidyverse
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## Loading required package: caret

## Warning: package 'caret' was built under R version 4.0.5

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

Analysis and methods

Data analysis

We explore data set to see its variable names and variable classes.

```
##         ph Hardness   Solids Chloramines  Sulfate Conductivity Organic_carbon
## 1       NA 204.8905 20791.32    7.300212 368.5164     564.3087      10.379783
## 2 3.716080 129.4229 18630.06    6.635246       NA     592.8854      15.180013
## 3 8.099124 224.2363 19909.54    9.275884       NA     418.6062      16.868637
## 4 8.316766 214.3734 22018.42    8.059332 356.8861     363.2665      18.436524
## 5 9.092223 181.1015 17978.99    6.546600 310.1357     398.4108      11.558279
## 6 5.584087 188.3133 28748.69    7.544869 326.6784     280.4679       8.399735
##   Trihalomethanes Turbidity Potability
## 1        86.99097  2.963135          0
## 2        56.32908  4.500656          0
## 3        66.42009  3.055934          0
## 4       100.34167  4.628771          0
## 5        31.99797  4.075075          0
## 6        54.91786  2.559708          0
```

Variable names:

```
##  [1] "ph"              "Hardness"        "Solids"          "Chloramines"
##  [5] "Sulfate"         "Conductivity"    "Organic_carbon"  "Trihalomethanes"
##  [9] "Turbidity"       "Potability"
```

Structure of the data:

```
str(maindata)
```

```
## 'data.frame':    3276 obs. of  10 variables:
##  $ ph             : num  NA 3.72 8.1 8.32 9.09 ...
##  $ Hardness       : num  205 129 224 214 181 ...
##  $ Solids         : num  20791 18630 19910 22018 17979 ...
##  $ Chloramines    : num  7.3 6.64 9.28 8.06 6.55 ...
##  $ Sulfate        : num  369 NA NA 357 310 ...
##  $ Conductivity   : num  564 593 419 363 398 ...
##  $ Organic_carbon : num  10.4 15.2 16.9 18.4 11.6 ...
##  $ Trihalomethanes: num  87 56.3 66.4 100.3 32 ...
##  $ Turbidity      : num  2.96 4.5 3.06 4.63 4.08 ...
##  $ Potability     : int  0 0 0 0 0 0 0 0 0 0 ...
```

Exploring if there are any missing values:

```
##              ph        Hardness          Solids     Chloramines         Sulfate
##             491               0               0               0             781
##    Conductivity  Organic_carbon Trihalomethanes       Turbidity      Potability
##               0               0             162               0               0
```

From the above results, we see that ph, sulfate, and Trihalomethanes columns have missing values, so we remove related rows.

Dropping missing values:

```
maindata <- na.omit(maindata)
```

Changing Potability columns values from 0 and 1 to "potable", and "non-potable" and making them as factor:

```
# Changing Potability columns values from 0 and 1 to "potable", and "non-potable" and making them as fa
data <- maindata %>% mutate(Potability=as.factor(ifelse(Potability==1, "potable","non-potable")))
```

Exploring the variables summary:

```
summary(data)
```

```
##        ph             Hardness          Solids         Chloramines
##  Min.   : 0.2275   Min.   : 73.49   Min.   :  320.9   Min.   : 1.391
##  1st Qu.: 6.0897   1st Qu.:176.74   1st Qu.:15615.7   1st Qu.: 6.139
##  Median : 7.0273   Median :197.19   Median :20933.5   Median : 7.144
##  Mean   : 7.0860   Mean   :195.97   Mean   :21917.4   Mean   : 7.134
```

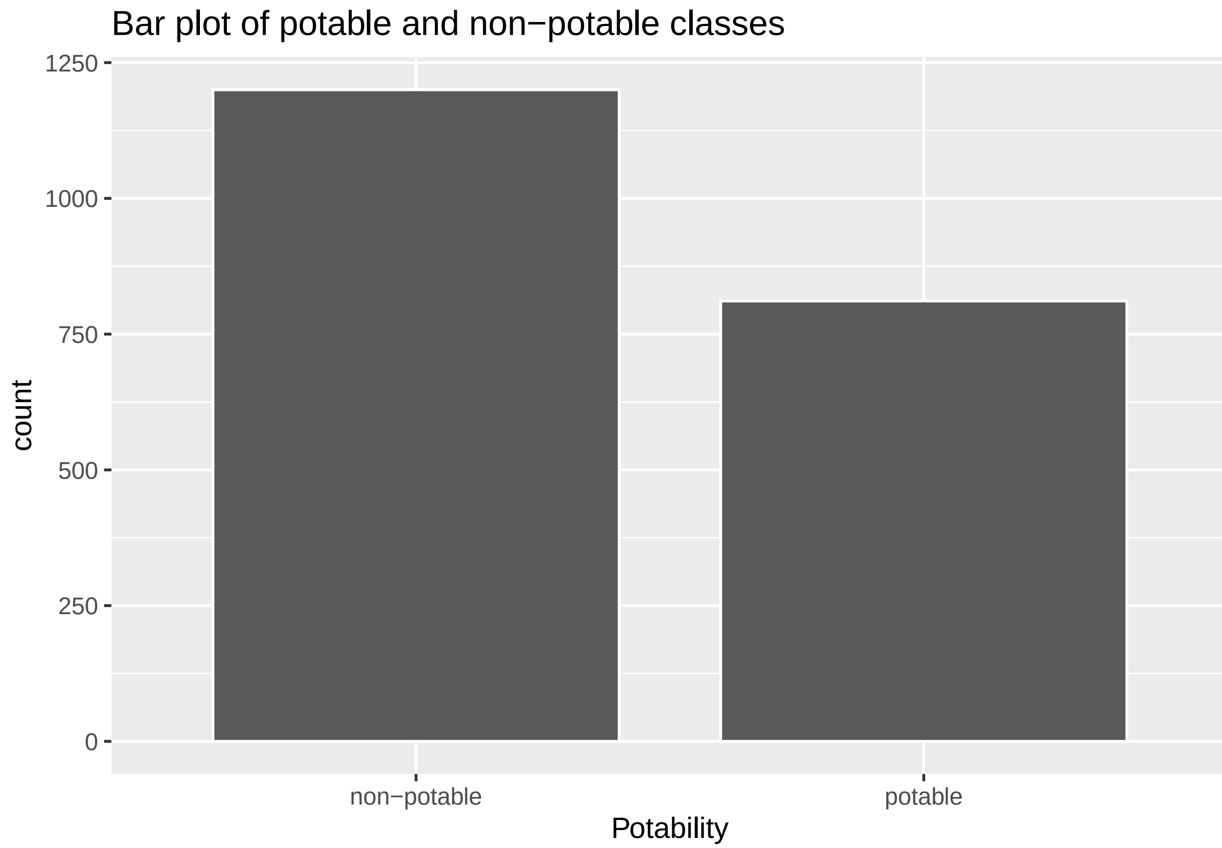
```
##  3rd Qu.: 8.0530   3rd Qu.:216.44   3rd Qu.:27182.6   3rd Qu.: 8.110
##  Max.   :14.0000   Max.   :317.34   Max.   :56488.7   Max.   :13.127
##     Sulfate       Conductivity   Organic_carbon  Trihalomethanes
##  Min.   :129.0   Min.   :201.6   Min.   : 2.20   Min.   :  8.577
##  1st Qu.:307.6   1st Qu.:366.7   1st Qu.:12.12   1st Qu.: 55.953
##  Median :332.2   Median :423.5   Median :14.32   Median : 66.542
##  Mean   :333.2   Mean   :426.5   Mean   :14.36   Mean   : 66.401
##  3rd Qu.:359.3   3rd Qu.:482.4   3rd Qu.:16.68   3rd Qu.: 77.292
##  Max.   :481.0   Max.   :753.3   Max.   :27.01   Max.   :124.000
##    Turbidity          Potability
##  Min.   :1.450   non-potable:1200
##  1st Qu.:3.443   potable    : 811
##  Median :3.968
##  Mean   :3.970
##  3rd Qu.:4.514
##  Max.   :6.495
```

Creating train and test data:

```
#Creating train and test data
set.seed(2007)
test_index <- createDataPartition(y = data$Potability, times = 1, p = 0.2, list = FALSE)
train <- data[-test_index,]
test <- data[test_index,]
```

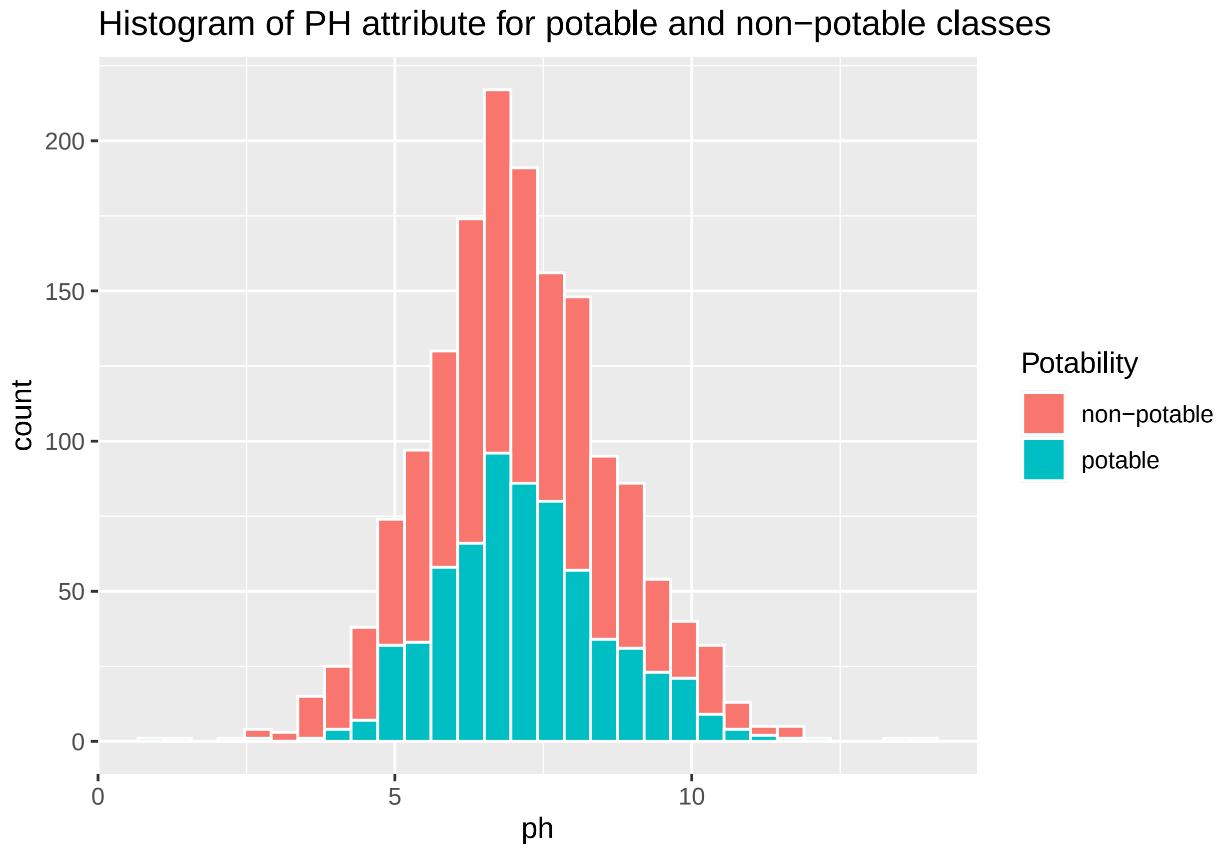
Data visualization

Bar plot of the data set to see the number of data in potable and non-potable classes



From the above bar plot, it is seen that we have more data in non-potable class than potable one. About 1100 in non-potable class, and 850 in potable class.

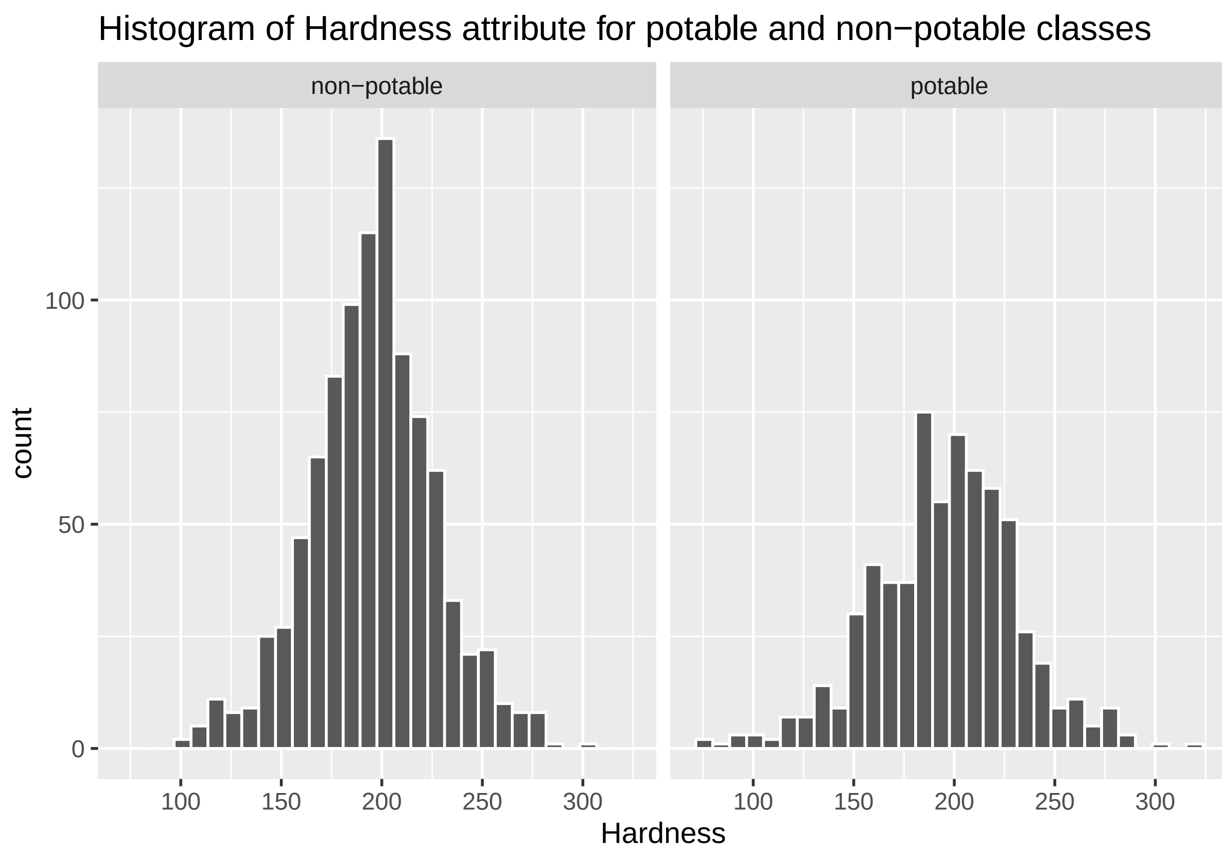
Exploring the distribution of “ph” variable for potable and non-potable classes:



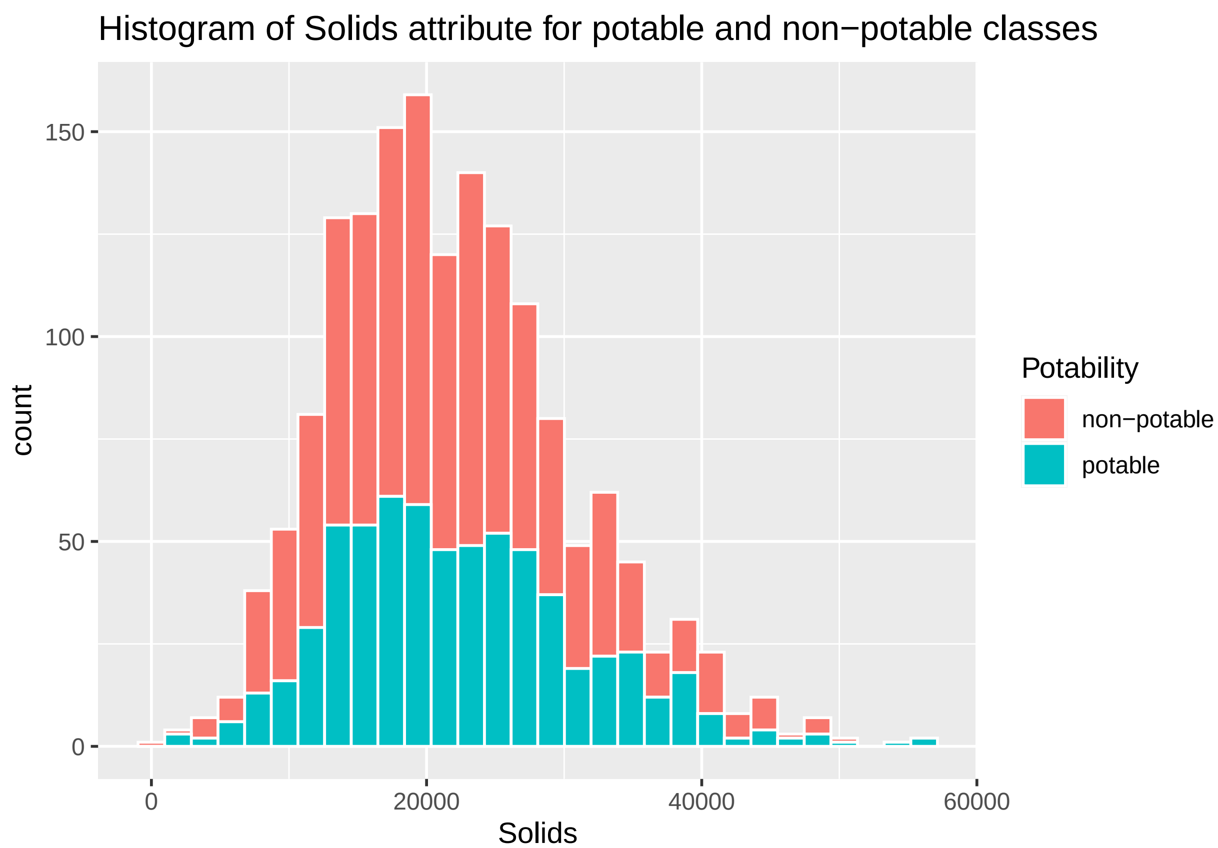
From the above graph we can state that both potable and non-potable classes have the same almost normal distribution with more data in non-potable class.

Now, we do the same for the other variables

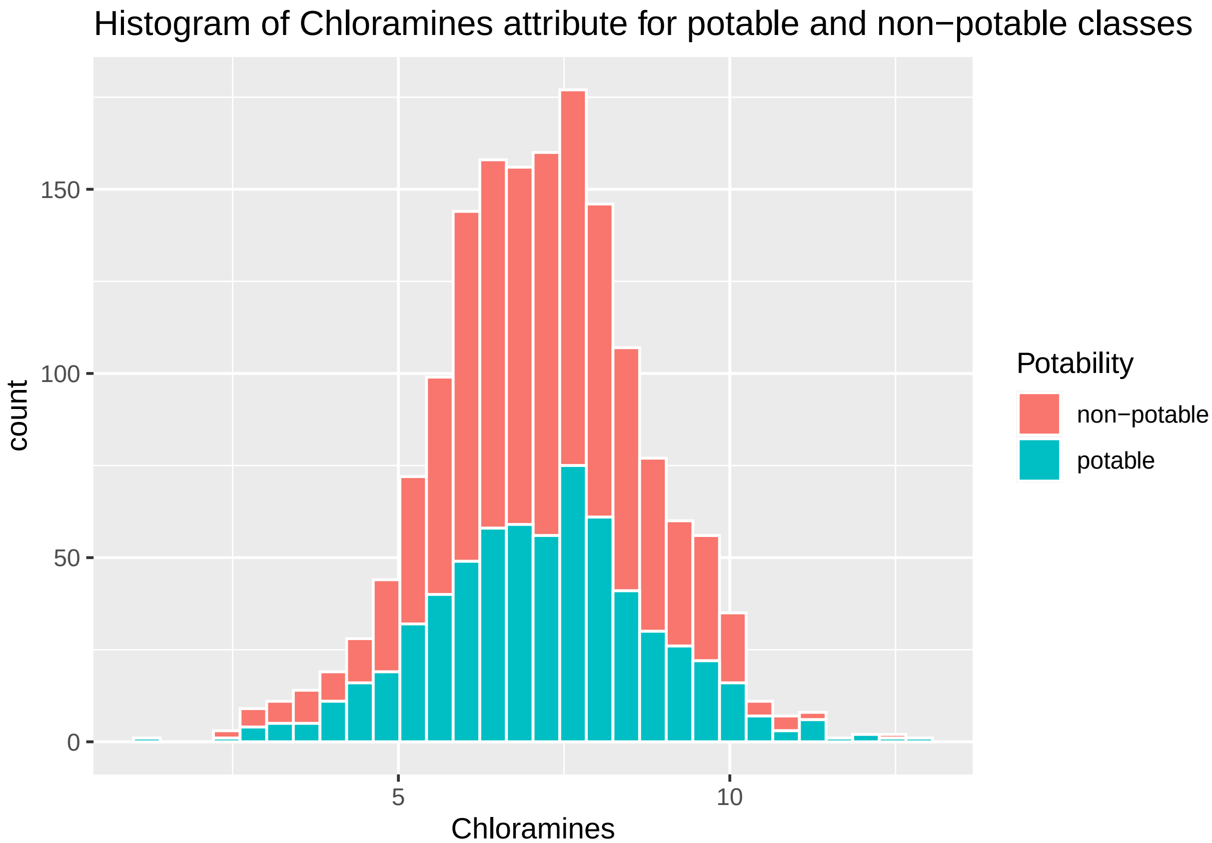
Hardness variable distribution:



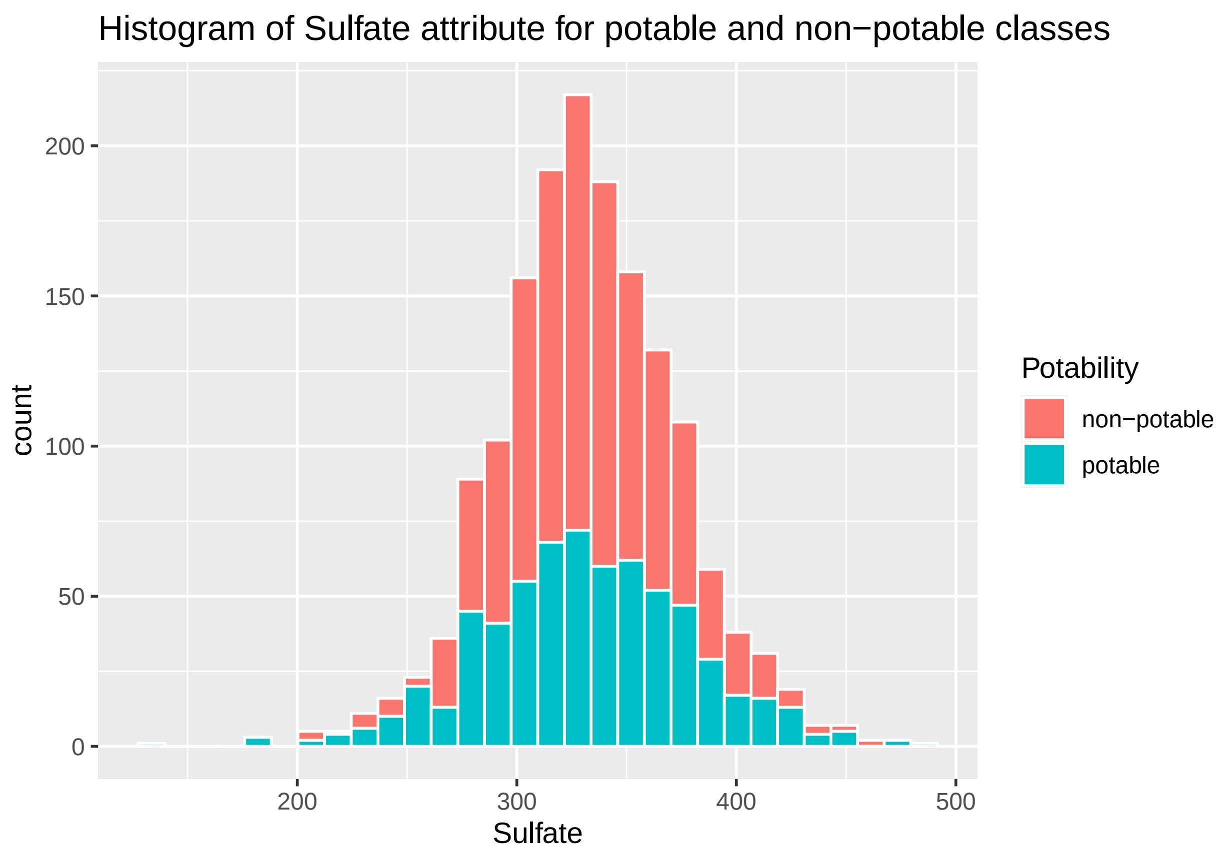
Solids variable distribution:



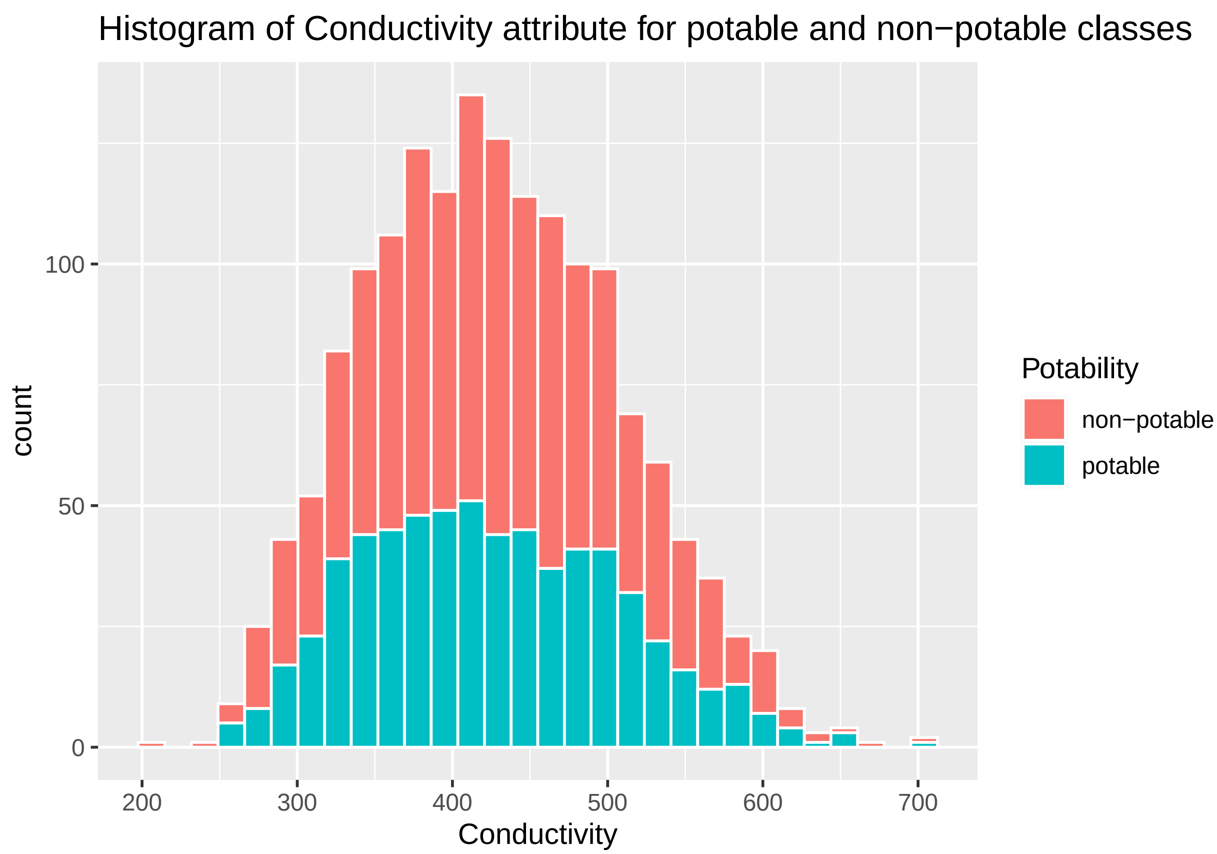
Chloramines variable distribution:



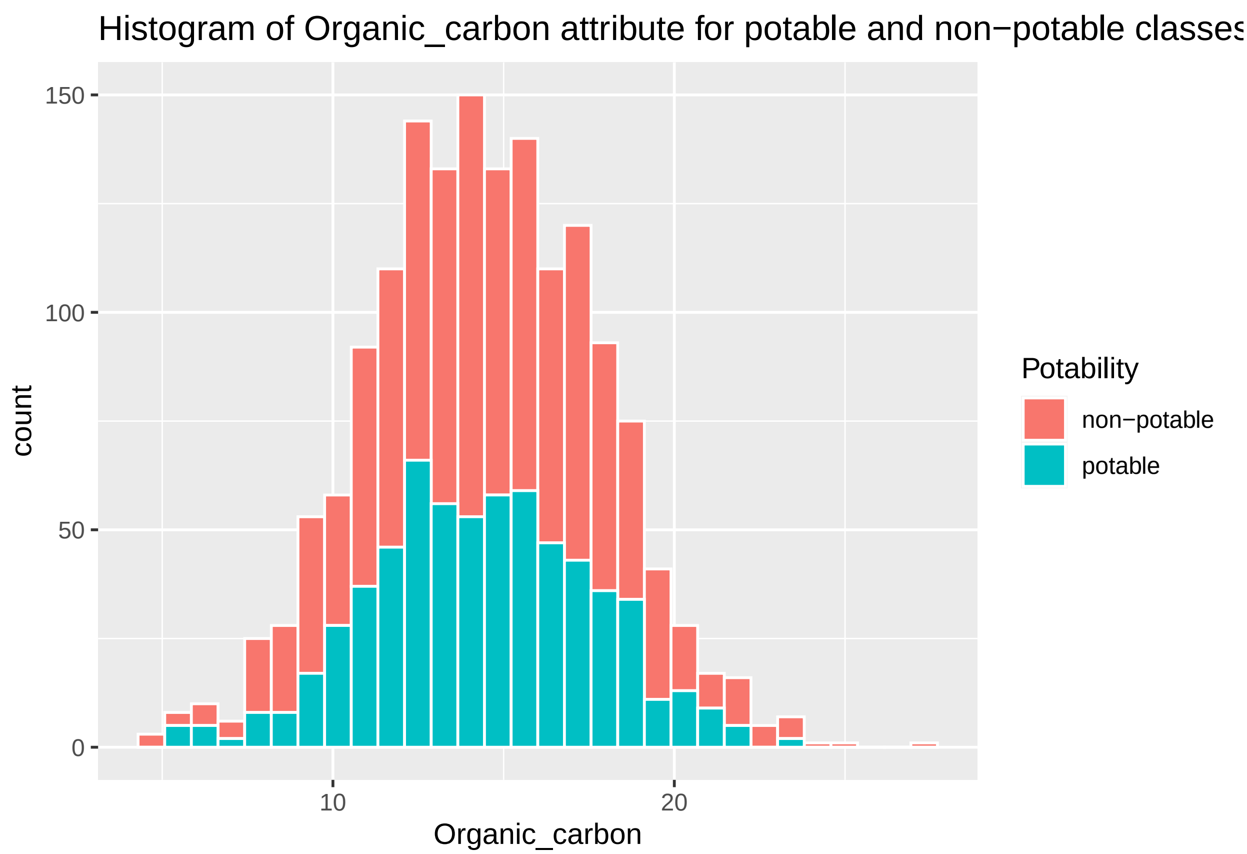
Sulfate variable distribution:



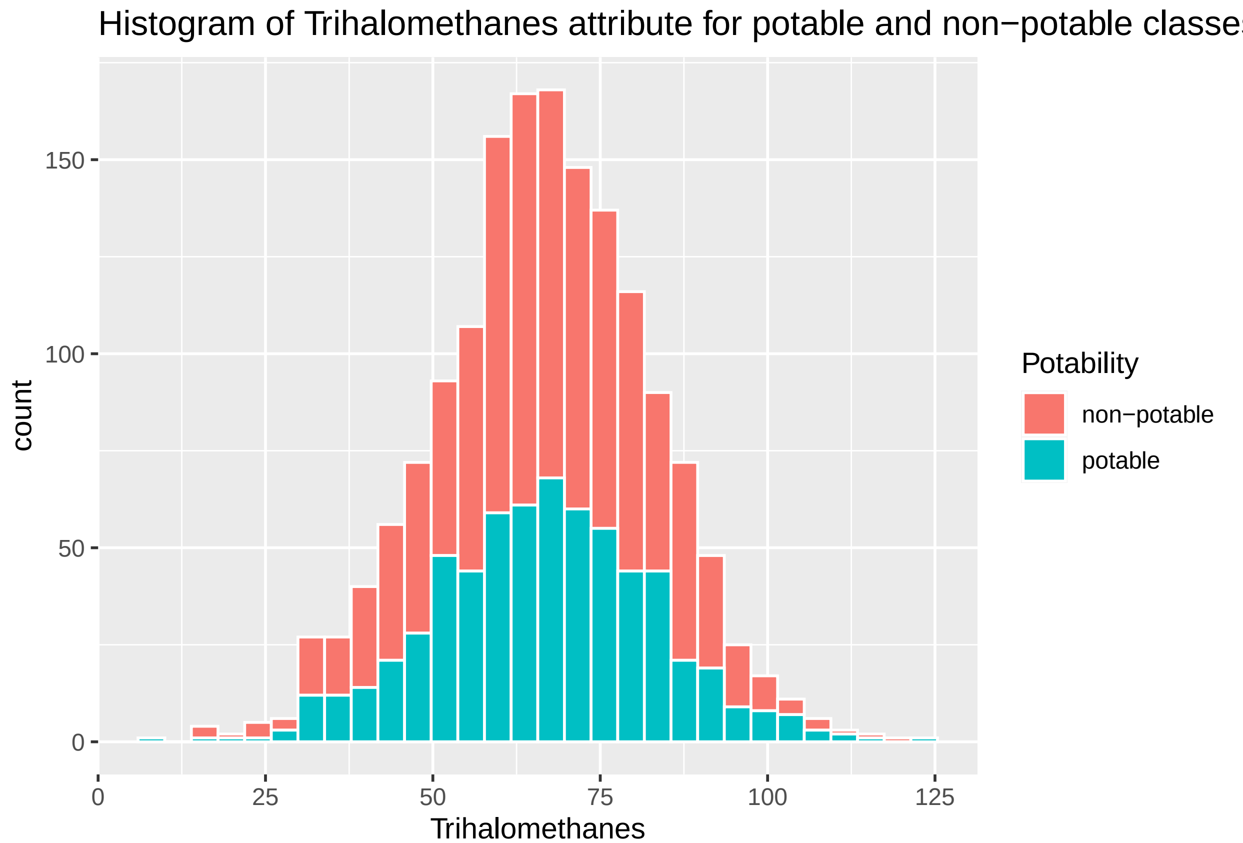
Conductivity variable distribution:



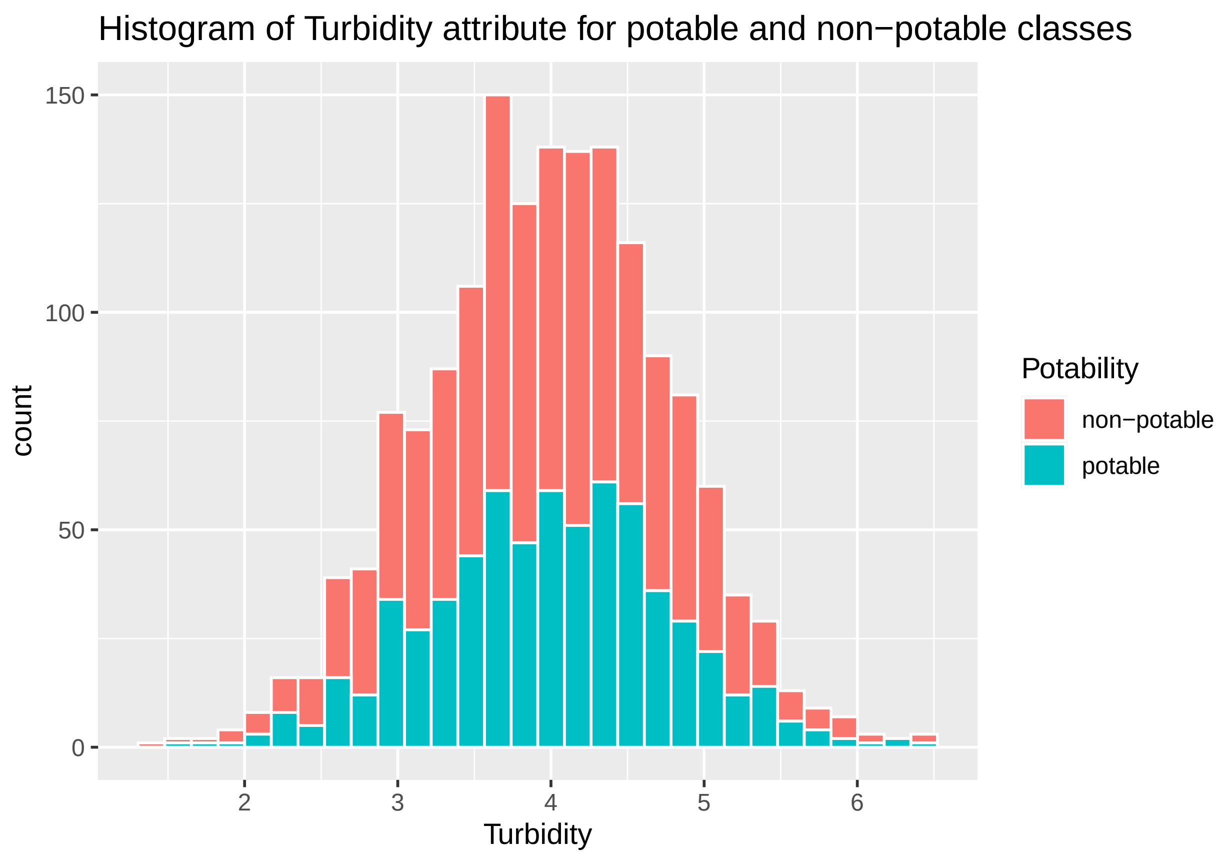
Organic_carbon variable distribution:



Trihalomethanes variable distribution:



Turbidity variable distribution:



Finding the correlation between variables

```
##                          ph    Hardness       Solids  Chloramines      Sulfate
## ph               1.00000000  0.10894811 -0.087614993 -0.024768491  0.010524348
## Hardness         0.10894811  1.00000000 -0.053268885 -0.022684975 -0.108520618
## Solids          -0.08761499 -0.05326888  1.000000000 -0.051789064 -0.162769204
## Chloramines     -0.02476849 -0.02268498 -0.051789064  1.000000000  0.006254057
## Sulfate          0.01052435 -0.10852062 -0.162769204  0.006254057  1.000000000
## Conductivity     0.01412785  0.01173055 -0.005197862 -0.028276649 -0.016192287
## Organic_carbon   0.02837522  0.01322386 -0.005484046 -0.023807630  0.026775563
## Trihalomethanes  0.01827788 -0.01540038 -0.015667788  0.014989930 -0.023346904
## Turbidity       -0.03584899 -0.03483094  0.019409428  0.013136570 -0.009933881
##                 Conductivity Organic_carbon Trihalomethanes    Turbidity
## ph               0.014127848    0.028375219     0.018277876 -0.035848994
## Hardness         0.011730548    0.013223861    -0.015400382 -0.034830942
## Solids          -0.005197862   -0.005484046    -0.015667788  0.019409428
## Chloramines     -0.028276649   -0.023807630     0.014989930  0.013136570
## Sulfate         -0.016192287    0.026775563    -0.023346904 -0.009933881
## Conductivity     1.000000000    0.015646727     0.004888475  0.012494892
## Organic_carbon   0.015646727    1.000000000    -0.005667486 -0.015428291
## Trihalomethanes  0.004888475   -0.005667486     1.000000000 -0.020497369
## Turbidity        0.012494892   -0.015428291    -0.020497369  1.000000000
```

If we consider 0.7 as a cutoff, there are no strong correlation between the variables.

Models

We will train 3 models which are Bayesian Generalized Linear, K-Nearest Neighbours, and Random Forest.

Bayesian Generalized Linear Model

The first model that I train is Bayesian Generalized Linear Model. I provide the confusion matrix results.

Confusion matrix results:

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-potable potable
##   non-potable         240     162
##   potable               0       1
##
##                Accuracy : 0.598
##                  95% CI : (0.5483, 0.6463)
##     No Information Rate : 0.5955
##     P-Value [Acc > NIR] : 0.4811
##
##                   Kappa : 0.0073
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.006135
##             Specificity : 1.000000
##          Pos Pred Value : 1.000000
##          Neg Pred Value : 0.597015
##              Prevalence : 0.404467
##          Detection Rate : 0.002481
##    Detection Prevalence : 0.002481
##       Balanced Accuracy : 0.503067
##
##        'Positive' Class : potable
##
```

K-Nearest Neighbours model

For the second model, I use K-Nearest Neighbours.

Confusion matrix results:

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-potable potable
##   non-potable         192      94
##   potable              48      69
##
##                Accuracy : 0.6476
##                  95% CI : (0.5988, 0.6943)
```

```
##     No Information Rate : 0.5955
##     P-Value [Acc > NIR] : 0.0181669
##
##                   Kappa : 0.2339
##
##  Mcnemar's Test P-Value : 0.0001592
##
##             Sensitivity : 0.4233
##             Specificity : 0.8000
##          Pos Pred Value : 0.5897
##          Neg Pred Value : 0.6713
##              Prevalence : 0.4045
##          Detection Rate : 0.1712
##    Detection Prevalence : 0.2903
##       Balanced Accuracy : 0.6117
##
##        'Positive' Class : potable
##
```

Random Forest model

The final model being trained is Random forest.

Confusion matrix results:

```
## Confusion Matrix and Statistics
##
##              Reference
## Prediction    non-potable potable
##   non-potable         206      89
##   potable              34      74
##
##                Accuracy : 0.6948
##                  95% CI : (0.6473, 0.7394)
##     No Information Rate : 0.5955
##     P-Value [Acc > NIR] : 2.317e-05
##
##                   Kappa : 0.3302
##
##  Mcnemar's Test P-Value : 1.122e-06
##
##             Sensitivity : 0.4540
##             Specificity : 0.8583
##          Pos Pred Value : 0.6852
##          Neg Pred Value : 0.6983
##              Prevalence : 0.4045
##          Detection Rate : 0.1836
##    Detection Prevalence : 0.2680
##       Balanced Accuracy : 0.6562
##
##        'Positive' Class : potable
##
```

Results

In the following table, I have provided the results obtained by 3 models in the previous section.

```r
# Table of models results
results <-results %>% arrange(Accuracy)
results %>% knitr::kable("pipe")
```

Method	Accuracy	Sensitivity	Specifity
Bayesian GLM	0.5980149	0.0061350	1.0000000
K-Nearest Neighbours	0.6476427	0.4233129	0.8000000
Random Forest	0.6947891	0.4539877	0.8583333

Based on the table, since the specificity of the all models are much higher than sensitivity, we can claim that the models do better in predicting non-potable water when water is actually non-potable, and there are many false negative results.

Conclusion

In the current project, we explored water quality data set as a classification problem. Several machine learning models were trained and were applied to predict potability of the water in the test data set. Based on the results, we can state that there is still room for improvements and other machine learning models may result in better predictions compared to these 3 applied models.