

|  |
| --- |
| **SCENARIO** |
| Recent public health data indicate a troubling increase in kidney disease rates within specific suburban areas, attracting significant attention from public health practitioners. Determined to uncover the root causes and identify actionable risk factors to address this issue, the public health team has embarked on a comprehensive study. They have collected patient records and relevant information on medical factors and water quality, as provided in the dataset. |

**Data Description:**

|  |  |
| --- | --- |
| **Variable** | **Description** |
| PatientID | Unique identifier of each patient |
| Age | Age of the individual |
| Gender | Gender of the individual |
| BloodPressure | Systolic blood pressure in mmHg |
| BloodSugar | Fasting blood sugar levels in mg/dL |
| Cholesterol | Total cholesterol level in mg/dL |
| BodyMassIndex | BMI, a measure of body fat based on height and weight |
| SmokingStatus | Smoking status of the individual [Never/ Former/ Current] |
| ElectricConductivity | Measurement of the water’s ability to conduct electricity, which can indicate contamination in μS/cm |
| pH | pH level of the water |
| DissolvedOxygen | Amount of oxygen dissolved in water in mg/L |
| Turbidity | Measure of water clarity in NTU |
| TotalDissolvedSolids | Measure of dissolved substances in water in mg/L |
| NitriteLevel | Nitrite concentration in water in mg/L |
| NitrateLevel | Nitrate concentration in water in mg/L |
| LeadConcentration | Lead concentration in water in mg/L |
| ArsenicConcentration | Arsenic concentration in water in mg/L |
| Humidity | Ambient humidity level in % |
| KidneyDisease | Presence or absence of kidney disease  0 – Absence of kidney disease  1 – Presence of kidney disease |

\* Please note that this is a simulated data generated to resemble the real-world data for the purpose of this assignment.

Consider the scenario described and the data set provided [***KidneyData.csv***] to answer the following questions.

1. Identify the target variable and clearly specify the research question.

|  |
| --- |
| Target variable: KidneyDisease  Research Question: With given medical factors and water quality, estimate the probability of a patient has a presence of kidney disease. |

1. Understand the data and perform the necessary data pre-processing. Clearly explain the steps taken.

|  |
| --- |
| [Write the steps taken here.]  Step 1: Add new feature *TdsEcRatio* = *TotalDissolvedSolids* / *ElectricConductivity*  Step 2: Remove *PatientID* and encode categorical features (*Gender*, *SmokingStatus*)  Step 3: Scale numerical features by subtracting the mean and then dividing by standard deviation  Step 4: Split train set (80%) and test set (20%) |
| Print the structure of the data before cleaning and pre-processing here. [*Hint: use str() function*]  'data.frame': 500 obs. of 19 variables:  $ PatientID : chr "TIW5219" "QLJ3151" "GRL2542" "WMM4122" ...  $ Age : int 120 10 58 22 52 53 76 45 57 30 ...  $ Gender : chr "Female" "Female" "Female" "Female" ...  $ BloodPressure : int 118 143 300 20 150 141 194 151 140 141 ...  $ BloodSugar : num 156 162 121 154 159 ...  $ Cholesterol : int 165 214 222 212 600 199 251 200 215 205 ...  $ BMI : num 31.7 23.9 16.3 21.9 23.8 18.3 26.2 22.2 19.5 25.7 ...  $ SmokingStatus : chr "Former" "Never" "Former" "Never" ...  $ ElectricConductivity: num 336 297 378 312 222 ...  $ pH : num 7.4 7.48 7.49 6.03 6.77 7.34 7.01 7.46 7.38 6.7 ...  $ DissolvedOxygen : num 9.57 8.49 8.18 7.35 7.4 8 9.79 8.72 8.04 6.98 ...  $ Turbidity : num 1.44 1.21 0.88 1.15 0.73 0.71 1.16 0.98 1.47 1.1 ...  $ TotalDissolvedSolids: num 455 423 434 400 349 ...  $ NitriteLevel : num 0.165 0.075 0.005 0.088 0.119 0.076 0.177 0.044 0.114 0.042 ...  $ NitrateLevel : num 1.97 1.74 1.4 0.88 0.71 1 1.13 1.13 1.13 0.82 ...  $ LeadConcentration : num 0.0099 0.012 0.0173 0.0133 0.0155 0.005 0.012 0.0106 ...  $ ArsenicConcentration: num 0.0063 0.0062 0.0092 0.0086 0.0011 0.009 0.0035 0.0062 ...  $ Humidity : num 48.7 65.3 93.2 67.4 43.3 57.6 50.8 70.5 55.6 72.9 ...  $ KidneyDisease : int 0 1 0 1 1 0 1 1 0 1 ... |
| Print the structure of the training data after cleaning and pre-processing here.  'data.frame': 400 obs. of 19 variables:  $ Age : num [1:400, 1] 1.131 0.346 -0.929 -0.978 -0.586 ...  $ Gender : num [1:400, 1] -0.922 1.082 -0.922 1.082 1.082 ...  $ BloodPressure : num [1:400, 1] -0.799 -0.223 -0.367 -0.271 -1.039 ...  $ BloodSugar : num [1:400, 1] 1.2031 0.094 1.0687 0.0716 0.0921 ...  $ Cholesterol : num [1:400, 1] -0.0723 0.6907 1.1015 0.2798 -0.131 ...  $ BMI : num [1:400, 1] -0.483 -1.284 0.528 1.33 -0.197 ...  $ SmokingStatus : num [1:400, 1] -1.2857 -1.2857 -0.0399 1.206 1.206 ...  $ ElectricConductivity: num [1:400, 1] -0.045 0.71 -0.257 1.093 -1.208 ...  $ pH : num [1:400, 1] 0.226 -0.995 -1.847 -0.278 -0.491 ...  $ DissolvedOxygen : num [1:400, 1] 1.104 -0.655 0.256 -0.231 -0.189 ...  $ Turbidity : num [1:400, 1] -0.128 -0.75 -0.559 1.789 1.022 ...  $ TotalDissolvedSolids: num [1:400, 1] -0.44 0.359 -0.104 1.153 -0.359 ...  $ NitriteLevel : num [1:400, 1] 0.545 1.665 -1.614 1.115 0.198 ...  $ NitrateLevel : num [1:400, 1] -0.38 0.575 1.388 -0.542 -0.237 ...  $ LeadConcentration : num [1:400, 1] 1.111 -0.0473 0.9636 -0.5949 -2.0903 ...  $ ArsenicConcentration: num [1:400, 1] -0.666 0.0169 -0.0319 0.9924 -0.0807 ...  $ Humidity : num [1:400, 1] 1.347 0.661 -0.733 -1.645 0.414 ...  $ KidneyDisease : Factor w/ 2 levels "0","1": 2 1 2 2 2 2 2 2 1 2 ...  $ TdsEcRatio : num [1:400, 1] -0.379 -0.711 0.111 -0.64 1.423 ... |

1. Perform a thorough data exploration using the provided dataset. You may use various visualization techniques (such as histograms, scatter plots, box plots, correlation matrices, etc.) to uncover significant patterns and insights. Interpret your outputs and discuss key findings. [Hint: You may use as many plots as necessary and make sure to interpret them.] **(10 Marks)**

|  |
| --- |
| The histograms of numerical features show that the data distribution shape of patients with and without Kidney disease are quiet similar. This implicates that the dataset is balanced between two groups. (see ***Figure 1***)  According to the boxplots, *ElectricConductivity, TotalDissolvedSolids, BloodSugar, DissolvedOxygen, Turbidity* and *TdsEcRatio* have significant difference in median between two groups. It implies high possibility to be able to classify instances into target groups based on these features. Besides that, the range of all variables are very different. So, we need to scale them (such as standardization or normalization) to avoid bias due to high variances. When it comes to outliers, there are still some outliers but not too much, so keeping them doesn’t matter. (see ***Figure 2***)  The correlation matrix shows that *ElectricConductivity* and *TotalDissolvedSolids* have a significant correlation. Therefore, it might be that we do not need both these variables in the model or can combine them to get a new feature enhancing the model accuracy. In this report, the new feature *TdsEcRatio* which is calculated by *TotalDissolvedSolids* dividing by *ElectricConductivity* has less correlation with *TotalDissolvedSolids* than *ElectricConductivity* and still have a significant correlation with *KidneyDesease*. It seems to be more suitable for model than *ElectricConductivity.* (see ***Figure 3***) |

1. Use logistic regression to answer the research question. Clearly explain the process or all the steps involved.

|  |
| --- |
| Step 1: Build logistic regression model to estimate *KidneyDisease* from all other features. Then, print model summary to pick features which has contributed significantly to the estimation. (see ***Snippet 1***)  Step 2: Define a function for evaluating the model accuracy (use confusion matrix) (see ***Snippet 2***)  Step 3: Build another model based on selected features and evaluate new model accuracy  Step 4: Print new model summary. With features which don’t have significant coefficients or have significant coefficients but at low level of confidence, get rid of them one by one and back to step 3. Then, compare the accuracy to the old model, if it decreases, keep the feature; otherwise, keep the model and try another one. (see ***Snippet 3, 4, 5, 6 & 7***)  Step 5: Choose the model with highest accuracy. |

1. Give your resultant model.

|  |
| --- |
| Call:  glm(formula = KidneyDisease ~ BloodPressure + DissolvedOxygen +  Turbidity + TotalDissolvedSolids + TdsEcRatio, family = binomial,  data = train)  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) 2.6943 0.2832 9.514 < 2e-16 \*\*\*  BloodPressure 0.4073 0.1503 2.710 0.006725 \*\*  DissolvedOxygen -0.4527 0.1786 -2.534 0.011263 \*  Turbidity 0.5804 0.1727 3.361 0.000776 \*\*\*  TotalDissolvedSolids -2.0546 0.2583 -7.956 1.78e-15 \*\*\*  TdsEcRatio 1.6129 0.2815 5.730 1.01e-08 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  (Dispersion parameter for binomial family taken to be 1)  Null deviance: 408.50 on 399 degrees of freedom  Residual deviance: 228.73 on 394 degrees of freedom  AIC: 240.73  Number of Fisher Scoring iterations: 6 |

--- End of questions for Part A. Part B will be available soon ---

**APPENDIX**

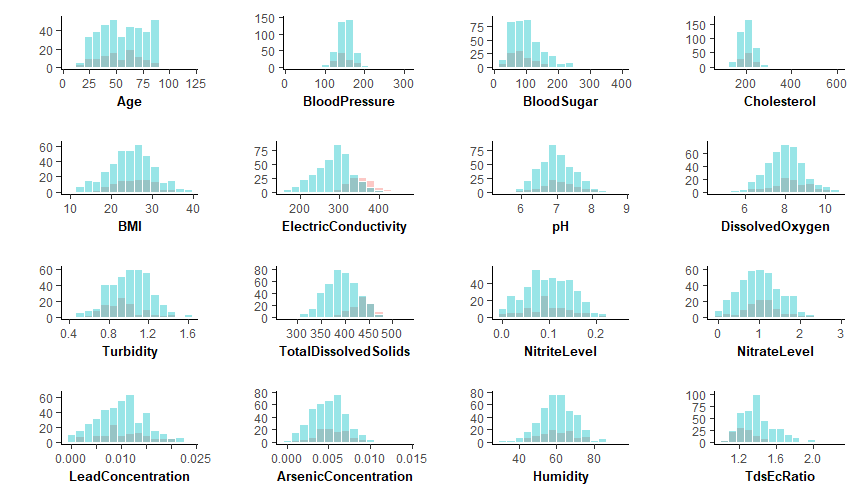


Figure : Histogram (S)

A chart of different types of numbers

Description automatically generated with medium confidence

Figure : Boxplot

A screenshot of a graph

Description automatically generated

Figure : Correlation Matrix

|  |  |
| --- | --- |
| [Code] | model=glm(KidneyDisease ~ ., data=train ,family=binomial)  summary(model) |
| Result |  |

Snippet : Model with all features

|  |  |
| --- | --- |
| [Code] | evaluate <- function(model, test.features, test.target) {  predicted <- as.numeric(predict (model, test.features, type="response")> .5)  xtab <- table(predicted, test.target)  cm <- caret::confusionMatrix(xtab)  return (cm)  } |

Snippet : Define evaluation function

|  |  |
| --- | --- |
| [Code] | model.select=glm(KidneyDisease ~ BloodPressure +  pH +  DissolvedOxygen +  Turbidity +  TotalDissolvedSolids +  NitriteLevel +  Humidity +  TdsEcRatio, data=train ,family=binomial)  summary.glm(model.select)  evaluate(model.select, test.features, test.target) |
| Result |  |

Snippet : model.select

|  |  |
| --- | --- |
| [Code] | model.select2=glm(KidneyDisease ~ BloodPressure +  pH +  DissolvedOxygen +  Turbidity +  TotalDissolvedSolids +  Humidity +  TdsEcRatio, data=train ,family=binomial)  summary.glm(model.select)  evaluate(model.select, test.features, test.target) |
| Result |  |

Snippet : model.select2

|  |  |
| --- | --- |
| [Code] | model.select3=glm(KidneyDisease ~ BloodPressure +  pH +  DissolvedOxygen +  Turbidity +  TotalDissolvedSolids +  TdsEcRatio, data=train ,family=binomial)  summary.glm(model.select)  evaluate(model.select, test.features, test.target) |
| Result |  |

Snippet : model.select3

|  |  |
| --- | --- |
| [Code] | model.select4=glm(KidneyDisease ~ BloodPressure +  ElectricConductivity +  pH +  DissolvedOxygen +  Turbidity +  TotalDissolvedSolids +  NitriteLevel +  Humidity +  TdsEcRatio, data=train ,family=binomial)  summary.glm(model.select)  evaluate(model.select, test.features, test.target) |
| Result |  |

Snippet : model.select4

|  |  |
| --- | --- |
| [Code] | model.select5=glm(KidneyDisease ~ BloodPressure +  DissolvedOxygen +  Turbidity +  TotalDissolvedSolids +  TdsEcRatio, data=train ,family=binomial)  summary.glm(model.select)  evaluate(model.select, test.features, test.target) |
| Result |  |

Snippet : model.select6

**Full R script:**

# Read data and discover some basic information

rawdata <- read.csv("KidneyData.csv")

attach(rawdata)

head(rawdata)

names(rawdata)

dim(rawdata)

sapply(rawdata,class)

str(rawdata)

summary(rawdata)

table(rawdata$KidneyDisease)

# Declare libraries used in this script

library(ggplot2)

library(dplyr)

library(corrplot)

library(boot)

library(gridExtra)

library(caret)

# Step 1: Add new feature TdsEcRatio = TotalDissolvedSolids / ElectricConductivity

data.add <- cbind(rawdata,

TdsEcRatio=rawdata$TotalDissolvedSolids /

rawdata$ElectricConductivity)

# Plot boxplot

par(mfrow=c(4,4))

for ( f in 1:16) {

boxplot(data.add[c(-19,-8,-3,-1)][f][data.add[19]==1],

data.add[c(-19,-8,-3,-1)][f][data.add[19]==0],

names = c("Yes", "No"),

main=colnames(data.add[c(-19,-8,-3,-1)])[f])

}

# Plot correlation matrix

cor\_matrix = cor(data.add[c(-8,-3,-1)])

par(mfrow=c(1,1))

corrplot(cor\_matrix, method = 'color',

addCoef.col = "#2C2C2C", # Add coefficient of correlation

number.cex = 0.5,

tl.col="black",

tl.srt=45,#Text label color and rotation

order = 'AOE', tl.cex=0.6, cl.cex = 0.7)

data.add$KidneyDisease = factor(data.add$KidneyDisease)

# Plot histogram

theme\_set(

theme\_classic()

)

p <- list()

i=0

for ( f in 1:17) {

if (f == 16)

{ next }

i = i + 1

p[[i]] <- ggplot(data.add[c(-8,-3,-1)],

aes\_string(x = colnames(data.add[c(-8,-3,-1)])[f])) +

aes(fill = KidneyDisease) +

scale\_x\_continuous(guide = guide\_axis(check.overlap = TRUE))+

geom\_histogram(color = "white", alpha=0.4,

position = "identity", bins=15) +

scale\_color\_manual(values = c("#00AFBB", "#E7B800")) +

labs(y="") +

theme(legend.position = "none",

axis.title.x = element\_text(size=8, face = "bold"),

axis.text = element\_text(size=7))

}

grid.arrange(grobs = lapply(p, "+",

theme(plot.margin=margin(10,10,10,10))), cols = 2)

# Step 2: Remove column PatientID and encode categorical columns (Gender, SmokingStatus)

data.encode = data.add[c(-1)]

data.encode$Gender <- as.integer(factor(data.encode$Gender))

data.encode$SmokingStatus <- as.integer(factor(data.encode$SmokingStatus))

# Step 3: Remove column PatientID and encode categorical columns (Gender, SmokingStatus)

data.scaled = data.encode

data.scaled <- data.scaled %>% mutate(across(where(is.numeric), scale))

# Step 4: Split train set (80%) and test set (20%)

tr.id=sample(1:nrow(data.scaled),nrow(data.scaled)\* 0.8)

train=data.scaled[tr.id,]

test=data.scaled[-tr.id,]

test.features = test[-18]

test.target = as.numeric(test[,18])-1

str(train)

# Logistic Regression Model

# Build model with all features

model=glm(KidneyDisease ~ ., data=train ,family=binomial)

summary(model)

# Define a function to evaluate model accuracy

evaluate <- function(model, test.features, test.target) {

predicted <- as.numeric(predict (model, test.features, type="response")> .5)

xtab <- table(predicted, test.target)

cm <- caret::confusionMatrix(xtab)

return (cm)

}

# Build model based on selected features which have significant coefficients

model.select=glm(KidneyDisease ~ BloodPressure +

pH +

DissolvedOxygen +

Turbidity +

TotalDissolvedSolids +

NitriteLevel +

Humidity +

TdsEcRatio, data=train ,family=binomial)

model.select2=glm(KidneyDisease ~ BloodPressure +

pH +

DissolvedOxygen +

Turbidity +

TotalDissolvedSolids +

Humidity +

TdsEcRatio, data=train ,family=binomial)

model.select3=glm(KidneyDisease ~ BloodPressure +

pH +

DissolvedOxygen +

Turbidity +

TotalDissolvedSolids +

TdsEcRatio, data=train ,family=binomial)

model.select4=glm(KidneyDisease ~ BloodPressure +

ElectricConductivity +

pH +

DissolvedOxygen +

Turbidity +

TotalDissolvedSolids +

NitriteLevel +

Humidity +

TdsEcRatio, data=train ,family=binomial)

model.select5=glm(KidneyDisease ~ BloodPressure +

DissolvedOxygen +

Turbidity +

TotalDissolvedSolids +

TdsEcRatio, data=train ,family=binomial)

# Evaluate model accuracy

summary.glm(model)

evaluate(model, test.features, test.target)

summary(model.select)

evaluate(model.select, test.features, test.target)

summary(model.select2)

evaluate(model.select2, test.features, test.target)

summary(model.select3)

evaluate(model.select3, test.features, test.target)

summary(model.select4)

evaluate(model.select4, test.features, test.target)

summary(model.select5)

evaluate(model.select5, test.features, test.target)