

|  |
| --- |
| **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 |

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

1. Build a logistic regression model incorporating polynomial terms. Clearly outline and explain each step of the process involved.

|  |
| --- |
| **Step 1**: Find the highest degree for each feature which will not cause overfitting when applied to polynomial logistic regression (“*Warning message: glm.fit: fitted probabilities numerically 0 or 1 occurred*” will be raised). The experiments (randomly start with d=5) showed that the appropriate degrees should be as follows:   * *Age*: 5, *Gender*: 1, *BloodPressure*: 2, *BloodSugar*: 5, *Cholesterol*: 1, *BMI*: 4, *SmokingStatus*: 2, *ElectricConductivity*: 2, *pH*: 5, *DissolvedOxygen*: 3, *TdsEcRatio*: 2, *NitriteLevel*: 3 * *NitrateLevel*, *LeadConcentration*, *Turbidity*, *TotalDissolvedSolids*, *ArsenicConcentration*, *Humidity* always made model overfitting. Therefore, they are considered to get rid of when building the model.   *Note:*  *Experiment implementation: Add features one by one into the model, start with d=5, decrease 1 in value of d whenever warning was raised. Keep the value of d without warning as the highest degree and continue the process.*  **Step 2**: Build model with selected features and degrees. Then choose the features which are significant to use in the final model. The result suggested *BloodPressure* (d=1), *ElectricConductivity* (d=2), *pH* (d=1), *DissolvedOxygen* (d=1), *TdsEcRatio* (d=1) as significant features *(*see ***Snippet 1****)*.  **Step 3**: Build model with selected features and measure the accuracy of model. Then, improve the accuracy by checking whether exists any feature which has negative effect on the model. The method applied is that get rid of feature one by one and measure the accuracy of new model. If the accuracy increases, it means that feature has negative effect on the model, and we will remove it. The results showed that *TdsEcRatio* (d=1) and *pH* (d=1) have negative effect and should be removed *(*see ***Snippet 2, 3, 4, 5, 6****)*.  **Step 4**: Choose the model with highest accuracy. |

1. Give the resultant accepted model (i.e. write the model equation) based on your findings above. Justify your answer clearly.

|  |
| --- |
| The resultant accepted model is:  The model summary shows that variables *ElectricConductivity*, *BloodPressure*, *DissolvedOxygen* have significant contribution to predict the likelihood of whether a patient have kidney disease or not.   * ***ElectricConductivity*** has negative contribution in degree 2 and positive in degree 1 at the significant level 0.001 (p-value are equal to 3.00e-05 and 2.81e-10 respectively). * ***BloodPressure*** has positive contribution in degree 1 at the significant level 0.05 (p-value= 0.01513). * ***DissolvedOxygen*** has negative contribution in degree 1 at the significant level 0.01 (p-value= 0.00707).   The model can predict accurately 88% in general, 92.11% in terms of negative class (patient with kidney disease) and 75% in terms of positive class (patient without kidney disease) *(*see ***Snippet 4****)*. |

1. Use decision tree model to answer the research question. Clearly outline and explain each step of the process involved.

|  |
| --- |
| **Step 1**: Create train and test set for decision tree model by using raw data without *PatientID* (because it is no meaningful in explaining the patient’s health information) and convert *KidneyDisease*, *Gender*, *SmokingStatus* feature to factor *(*see ***Snippet 7****)*.  **Step 2**: Build decision tree model with all features. The output model has 14 terminal nodes and actually uses 9 variables including *ElectricConductivity*, *TotalDissolvedSolids*, *pH*, *Age*, *BloodSugar*, *Turbidity*, *Cholesterol*, *BloodPressure* and *ArsenicConcentration* *(*see ***Snippet 8****)*.  **Step 3**: Plot the tree model and evaluate the misclassification rate. The misclassification rate when predicting for the test set is 0.06 *(*see ***Snippet 9, 10****)*.  **Step 4**: Use cross-validation to find the best size of tree which means lowest deviance. The result shows that the best size of tree is 5 *(*see ***Snippet 11****)*.  **Step 5**: Prune the model with the best size *(*see ***Snippet 12****)*.  **Step 6**: Plot the tree model and evaluate the misclassification rate. The misclassification rate after pruning is 0.05 *(*see ***Snippet 13****)*. |

1. Give the resultant model and interpret it. Clearly describe the terminal nodes [i.e. list the profiles]. *[Include the relevant R output]*

|  |
| --- |
| The resultant model is as follows:    This model fitted 400 observations with the ratio of patients without and with kidney disease 20.75% and 79.25% respectively. It uses 4 variables to classify new observations including *ElectricConductivity*, *Turbidity*, *Cholesterol* and *BloodPressure*. It has 5 terminal nodes as following:   * **ElectricConductivity < 320.95**: patients with *ElectricConductivity < 320.95* will be classified into *presence of kidney disease* (p=95.572%). * **ElectricConductivity > 320.95 & Turbidity < 1.005 & Cholesterol < 251.5**: patients satisfied these criteria will be classified into *absence of kidney disease* (p=85.484%). * **ElectricConductivity > 320.95 & Turbidity < 1.005 & Cholesterol > 251.5**: patients satisfied these criteria will be classified into *presence of kidney disease* (p=100%). * **ElectricConductivity > 320.95 & Turbidity > 1.005 & BloodPressure < 140**: patients satisfied these criteria will be classified into *absence of kidney disease* (p=100%). * **ElectricConductivity > 320.95 & Turbidity > 1.005 & BloodPressure > 140**: patients satisfied these criteria will be classified into *presence of kidney disease* (p=100%).   The misclassification rate of the model when predicting for the test set is 0.05 *(*see ***Snippet 13****)* |

1. Compare the different resultant models (Part A Question 5, Part B Question 2 and Question 4) you obtained above.

|  |
| --- |
| **The difference in using variables:**   * The logistic regression *(lm)* model uses variables *BloodPressure*, *DissolvedOxygen*, *Turbidity*, *TotalDissolvedSolids* and *TdsEcRatio.* * The polynomial logistic regression *(poly)* model uses variables *ElectricConductivity*, *BloodPressure*, *DissolvedOxygen*. * The decision tree *(tree)* model uses variables *ElectricConductivity*, *Turbidity*, *Cholesterol* and *BloodPressure*.   **The contribution level of variables**:   * In *lm* model, the most contributed variable is *TotalDissolvedSolids,* then *TdsEcRatio, Turbidity, DissolvedOxygen* and *BloodPressure.* * In *poly* model, the most contributed variable is *ElectricConductivity,* then *DissolvedOxygen* and *BloodPressure.* * In *tree* model, the top node is *ElectricConductivity,* then *Turbidity, Cholesterol* and *BloodPressure.*   **Accuracy**: With the same train and test set, *tree* model achieved highest accuracy score (0.95), then *lm* model (0.93) and *poly* model (0.88).  **Positive and negative classification**: All models tend to predict better in negative class (*TNRlm=0.9481* > *TPRlm=0.8696*, *TNRpoly=0.9211* > *TPRpoly=0.75*, *TNRtree=0.9733* > *TPRtree=0.88*) and *tree* model is the most balance in classification (balance accuracy 0.9386, 0.8969 and 0.8355 for *tree*, *lm* and *poly* respectively). |

1. Give the final accepted model based on your findings above and Part A. Justify your answer.

|  |
| --- |
| The final accepted model is the decision tree model (see Question 4) because this model achieved highest accuracy score (0.95, compared to 0.93 and 0.88 of the remaining models) and the most balance in classification (balance accuracy 0.9386, compared to 0. 8969 and 0.8355 of the remaining models). |

1. Apply an unsupervised learning technique of your choice to identify any interesting or hidden patterns in the dataset. Provide a clear explanation of the technique used and thoroughly describe your findings.

|  |
| --- |
| Apply the agglomerative hierarchical clustering technique whose idea is to gradually merge clusters together to get a hierarchy of cluster solutions, implemented *(*see ***Snippet 14*** for R code*)* as follows:  **Step 1**: Apply data pre-processing including encoding and normalizing in order to convert raw data into numerical and scaled data. In this work, we use the processed data in Part A again.  **Step 2**: Use function *dist* to calculate the distances between clusters.  **Step 3**: Apply function *hclust* to do the agglomerative hierarchical clustering with distances from step 2 and method “*ward*.*D2*”. We chose method “*ward*.*D2*” because this method can produce groups which are more well-separate and balance than the one produced by other methods (“single”, “average” and “complete”).  **Step 4**: Plot the cluster dendrogram.  **Step 5**: Based on the dendrogram, select the desired number of groups. In this work, select k=9.  **Step 6**: Use function *cutree* to extract cluster membership.  **Findings:**  The groups divided by this method have significant number of members except group 3:    The number of members with and without kidney disease in each group as following:    In general, group 5 and 8 can be efficient in classifying the presence of kidney disease. The other groups are not good compared to the models built so far. |

1. What are your conclusion and recommendations for this problem?

|  |
| --- |
| **Conclusion:**  It seems that the kidney disease rate is highly affected by the water quality with the occurrence of correlations between kidney disease and the water’s ability to conduct electricity or water contamination (*ElectricConductivity*, *TotalDissolvedSolids, TdsEcRatio),* water clarity in NTU *(Turbidity)* and amount of oxygen dissolved in water (*DissolvedOxygen).* The systolic blood pressure (*BloodPressure)* and total cholesterol level (*Cholesterol)* also have effects on this rate. By using resultant model in Question 6, we can predict whether a patient has the presence of kidney disease or not with about 95% accuracy.  **Recommendations:**   * To reduce the kidney disease rate, governments, public health practitioners and people should have actions and policies aimed to improve the water’s ability to conduct electricity, the water clarity and the amount of oxygen dissolved in water. Try to control the *ElectricConductivity* ratio greater than **320.95.** For examples, adding plants to the water and removing any decaying plants; implementing or improving the water filtration systems. * In terms of medical factors, try to keep the systolic blood pressure less than **140** and the total cholesterol level less than **251.5**. Otherwise, even with high-quality water, you also have high possibility of kidney disease. |

--- End of questions ---

APPENDIX

[Attach all your R codes and outputs here.]

|  |  |
| --- | --- |
| [Code] | d=5  model.poly.all = glm(KidneyDisease ~ poly(Age,d)  + poly(Gender,1)  + poly(BloodPressure,2)  + poly(BloodSugar,d)  + poly(Cholesterol,1)  + poly(BMI,4)  + poly(SmokingStatus,2)  + poly(ElectricConductivity,2)  + poly(pH,d)  + poly(DissolvedOxygen,3)  + poly(TdsEcRatio,2)  + poly(NitriteLevel,3)  , data=train ,family=binomial)  summary.glm(model.poly.all) |
| Result |  |

Snippet : The polynomial model with all features

|  |  |
| --- | --- |
| [Code] | model.poly.filter = glm(KidneyDisease ~  poly(BloodPressure,1,raw=TRUE)  + poly(ElectricConductivity,2,raw=TRUE)  + poly(pH,1,raw=TRUE)  + poly(DissolvedOxygen,1,raw=TRUE)  + poly(TdsEcRatio,1,raw=TRUE)  , data=train ,family=binomial)  summary.glm(model.poly.filter)  evaluate(model.poly.filter, test.features, test.target) |
| Result |  |

Snippet : The polynomial model with significant features

|  |  |
| --- | --- |
| [Code] | model.poly.filter1 = glm(KidneyDisease ~  poly(BloodPressure,1,raw=TRUE)  + poly(ElectricConductivity,2,raw=TRUE)  + poly(DissolvedOxygen,1,raw=TRUE)  + poly(TdsEcRatio,1,raw=TRUE)  , data=train ,family=binomial)  summary.glm(model.poly.filter1)  evaluate(model.poly.filter1, test.features, test.target) |
| Result |  |

Snippet : Remove feature pH

|  |  |
| --- | --- |
| [Code] | model.poly.filter2 = glm(KidneyDisease ~  poly(BloodPressure,1,raw=TRUE)  + poly(ElectricConductivity,2,raw=TRUE)  + poly(DissolvedOxygen,1,raw=TRUE)  , data=train ,family=binomial)  summary.glm(model.poly.filter2)  evaluate(model.poly.filter2, test.features, test.target) |
| Result |  |

Snippet : Remove TdsEcRatio (d=1)

|  |  |
| --- | --- |
| [Code] | model.poly.filter3 = glm(KidneyDisease ~  poly(BloodPressure,1,raw=TRUE)  + poly(ElectricConductivity,2,raw=TRUE)  , data=train ,family=binomial)  summary.glm(model.poly.filter3)  evaluate(model.poly.filter3, test.features, test.target) |
| Result |  |

Snippet : Remove DissolvedOxygen (d=1)

|  |  |
| --- | --- |
| [Code] | model.poly.filter4 = glm(KidneyDisease ~  poly(ElectricConductivity,2,raw=TRUE)  + poly(DissolvedOxygen,1,raw=TRUE)  , data=train ,family=binomial)  summary.glm(model.poly.filter4)  evaluate(model.poly.filter4, test.features, test.target) |
| Result |  |

Snippet : Remove BloodPressure (d=1)

|  |  |
| --- | --- |
| [Code] | # Data pre-processing for decision tree model  data.tree = rawdata[c(-1)]  data.tree$KidneyDisease=ifelse(data.tree$KidneyDisease==0,'N','Y')  data.tree$KidneyDisease=as.factor(data.tree$KidneyDisease)  data.tree$Gender=as.factor(data.tree$Gender)  data.tree$SmokingStatus=as.factor(data.tree$SmokingStatus)  # Split train and test set  train.tree=data.tree[tr.id,]  test.tree=data.tree[-tr.id,]  test.tree.features = test.tree[-18]  test.tree.target = test.tree[,18]  # Check dim and type of data after pre-processing  sapply(data.tree,class)  dim(train.tree)  dim(test.tree) |
| Result |  |

Snippet : Data pre-processing for decision tree model

|  |  |
| --- | --- |
| [Code] | # Build tree model with all features  tree\_model = tree(KidneyDisease~.,train.tree)  summary(tree\_model)dim(train.tree) |
| Result |  |

Snippet : Tree model with all features

|  |  |
| --- | --- |
| [Code] | # Plot tree  plot(tree\_model)  text(tree\_model,pretty = 0, cex=0.5) |
| Result |  |

Snippet : Plot the tree

|  |  |
| --- | --- |
| [Code] | # Evaluate the model  tree\_pred = predict(tree\_model,test.tree.features,type="class")  table(tree\_pred,test.tree.target)  tab1 <- table(tree\_pred,test.tree.target)  MissclassificationRate <- (tab1[1,2]+tab1[2,1])/sum(tab1)  MissclassificationRate |
| Result |  |

Snippet : Evaluate the tree model

|  |  |
| --- | --- |
| [Code] | # Find the best size of tree  cv.tree\_model=cv.tree(tree\_model,FUN=prune.misclass)  names(cv.tree\_model)  plot(cv.tree\_model$size, cv.tree\_model$dev, type = "b") |
| Result |  |

Snippet : The correlation between size and deviance of tree model

|  |  |
| --- | --- |
| [Code] | # Prune tree model  prune.tree=prune.misclass(tree\_model,best=5)  plot(prune.tree)  text(prune.tree,pretty=0, cex=0.5) |
| Result |  |

Snippet : Pruning the tree model with size = 5

|  |  |
| --- | --- |
| [Code] | # Evaluate the pruned model  tree\_pred = predict(prune.tree,test.tree.features,type="class")  table(tree\_pred,test.tree.target)  tab1 <- table(tree\_pred,test.tree.target)  MissclassificationRate <- (tab1[1,2]+tab1[2,1])/sum(tab1)  MissclassificationRate  caret::confusionMatrix(tab1) |
| Result |  |

Snippet : Evaluate the pruned model

|  |  |
| --- | --- |
| [Code] | hh = hclust(dist(data.scaled[,-18]), method="ward.D2")  plot(hh, xlab=" ", sub = paste("Complete"," link cluster analysis"))  rect.hclust(hh, k=9)  cut\_avg <- cutree(hh, k=9)  seeds\_df\_cl <- mutate(data.scaled, cluster = cut\_avg)  df1 <- seeds\_df\_cl %>% count(cluster) %>% rename(n = n)  cluster.nkd <- seeds\_df\_cl %>% filter(KidneyDisease==0)  df2 <- cluster.nkd %>% count(cluster) %>% rename(WithoutKD = n)  cluster.kd <- seeds\_df\_cl %>% filter(KidneyDisease==1)  df3 <- cluster.kd %>% count(cluster) %>% rename(KD = n)  df\_list <- list(df1, df2, df3)  df <- Reduce(function(x, y) merge(x, y, all=TRUE), df\_list)  df <- replace\_na(df,list(n=0,WithoutKD=0,KD=0))  df$KD.rate <- df$KD / df$n  df  mean(df$KD.rate) |
| Result |  |

Snippet : Unsupervised clustering implementation

**Full R script Part B:**

#================= PART B HERE=====================#

# Build model with all features

d=5

model.poly.all= glm(KidneyDisease ~ poly(Age,d)

+ poly(Gender,1)

+ poly(BloodPressure,2)

+ poly(BloodSugar,d)

+ poly(Cholesterol,1)

+ poly(BMI,4)

+ poly(SmokingStatus,2)

+ poly(ElectricConductivity,2)

+ poly(pH,d)

+ poly(DissolvedOxygen,3)

+ poly(TdsEcRatio,2)

+ poly(NitriteLevel,3)

, data=train ,family=binomial)

summary.glm(model.poly.all)

# Build model with significant features and measure accuracy

model.poly.filter = glm(KidneyDisease ~

poly(BloodPressure,1,raw=TRUE)

+ poly(ElectricConductivity,2,raw=TRUE)

+ poly(pH,1,raw=TRUE)

+ poly(DissolvedOxygen,1,raw=TRUE)

+ poly(TdsEcRatio,1,raw=TRUE)

, data=train ,family=binomial)

summary.glm(model.poly.filter)

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

# Reducing features and measure accuracy

model.poly.filter1 = glm(KidneyDisease ~

poly(BloodPressure,1,raw=TRUE)

+ poly(ElectricConductivity,2,raw=TRUE)

+ poly(DissolvedOxygen,1,raw=TRUE)

+ poly(TdsEcRatio,1,raw=TRUE)

, data=train ,family=binomial)

summary.glm(model.poly.filter1)

evaluate(model.poly.filter1, test.features, test.target)

model.poly.filter2 = glm(KidneyDisease ~

poly(BloodPressure,1,raw=TRUE)

+ poly(ElectricConductivity,2,raw=TRUE)

+ poly(DissolvedOxygen,1,raw=TRUE)

, data=train ,family=binomial)

summary.glm(model.poly.filter2)

evaluate(model.poly.filter2, test.features, test.target)

model.poly.filter3 = glm(KidneyDisease ~

poly(BloodPressure,1,raw=TRUE)

+ poly(ElectricConductivity,2,raw=TRUE)

, data=train ,family=binomial)

summary.glm(model.poly.filter3)

evaluate(model.poly.filter3, test.features, test.target)

model.poly.filter4 = glm(KidneyDisease ~

poly(ElectricConductivity,2,raw=TRUE)

+ poly(DissolvedOxygen,1,raw=TRUE)

, data=train ,family=binomial)

summary.glm(model.poly.filter4)

evaluate(model.poly.filter4, test.features, test.target)

#---------------------DONE poly------

library(tree)

# Data pre-processing for decision tree model

data.tree = rawdata[c(-1)]

data.tree$KidneyDisease=ifelse(data.tree$KidneyDisease==0,'N','Y')

data.tree$KidneyDisease=as.factor(data.tree$KidneyDisease)

data.tree$Gender=as.factor(data.tree$Gender)

data.tree$SmokingStatus=as.factor(data.tree$SmokingStatus)

# Split train and test set

train.tree=data.tree[tr.id,]

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

test.tree.features = test.tree[-18]

test.tree.target = test.tree[,18]

# Check dim and type of data after pre-processing

sapply(data.tree,class)

dim(train.tree)

dim(test.tree)

# Build tree model with all features

tree\_model = tree(KidneyDisease~.,train.tree)

summary(tree\_model)

# Plot tree

plot(tree\_model)

text(tree\_model,pretty = 0, cex=0.5)

# Evaluate the model

tree\_pred = predict(tree\_model,test.tree.features,type="class")

table(tree\_pred,test.tree.target)

tab1 <- table(tree\_pred,test.tree.target)

MissclassificationRate <- (tab1[1,2]+tab1[2,1])/sum(tab1)

MissclassificationRate

# Find the best size of tree

cv.tree\_model=cv.tree(tree\_model,FUN=prune.misclass)

names(cv.tree\_model)

plot(cv.tree\_model$size, cv.tree\_model$dev, type = "b")

# Prune tree model

prune.tree=prune.misclass(tree\_model,best=5)

plot(prune.tree)

text(prune.tree,pretty=0, cex=0.5)

# Evaluate the pruned model

tree\_pred = predict(prune.tree,test.tree.features,type="class")

table(tree\_pred,test.tree.target)

tab1 <- table(tree\_pred,test.tree.target)

MissclassificationRate <- (tab1[1,2]+tab1[2,1])/sum(tab1)

MissclassificationRate

caret::confusionMatrix(tab1)

print(prune.tree)

#-----------DONE tree--------------

# Implementing clustering

hh = hclust(dist(data.scaled[,-18]), method="ward.D2")

plot(hh, xlab=" ", sub = paste("Complete"," link cluster analysis"))

rect.hclust(hh, k=9)

# Extract membership

cut\_avg <- cutree(hh, k=9)

# Merge cluster result with dataset

seeds\_df\_cl <- mutate(data.scaled, cluster = cut\_avg)

# Pivot cluster and kidney disease class

df1 <- seeds\_df\_cl %>% count(cluster) %>% rename(n = n)

cluster.nkd <- seeds\_df\_cl %>% filter(KidneyDisease==0)

df2 <- cluster.nkd %>% count(cluster) %>% rename(WithoutKD = n)

cluster.kd <- seeds\_df\_cl %>% filter(KidneyDisease==1)

df3 <- cluster.kd %>% count(cluster) %>% rename(KD = n)

df\_list <- list(df1, df2, df3)

df <- Reduce(function(x, y) merge(x, y, all=TRUE), df\_list)

df <- replace\_na(df,list(n=0,WithoutKD=0,KD=0))

df$KD.rate <- df$KD / df$n

df

mean(df$KD.rate)