

**School of Computer, Data and Mathematical Sciences**

**COMP 7006 Data Science**

**Computer Based Assignment – PART B**

**Spring, 2024**

|  |  |
| --- | --- |
| Complete your details in this section. | |
| STUDENT ID: | 22119429 |
| STUDENT FIRST NAME: | Aimaier |
| STUDENT SURNAME: | Xirezati |

|  |  |
| --- | --- |
| QUESTIONS FORMAT: | Word processed document in PDF format; logically presenting answers to each question incorporating R outputs including graphs and charts. |
| TOTAL MARKS: | **60 Marks** |
| UNIT CO-ORDINATOR: | Dr. Liwan Liyanage |
| TUTOR: | Ms. Prathayne Nanthakumaran |
| TOTAL PAGES: | 39 |

|  |
| --- |
| **INSTRUCTIONS**  Please note that you are expected to answer the questions clearly in this document. Use the template included where relevant to answer. Give the R outputs, comments, and discussion clearly and logically in the rectangular box provided under each question. Attach all the R commands in the Appendix. Write the resulting **model equation** to the relevant questions. Once completed submit the answer scripts as a **PDF** via TurnItin link within vUWS site.  Please note that **10 Marks** are allocated for organization, reasoning, logical flow, and the inclusion of all correct R codes and outputs in the Appendix for both Part A and Part B. |

|  |
| --- |
| **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.

Consider the scenario described, the data set provided and your answers in Part A to answer the following questions.

1. Build a logistic regression model incorporating polynomial terms. Clearly outline and explain each step of the process involved. [*This question is designed to assess your critical thinking and analytical skills. Please note that guidance on how to complete the task will not be provided.*] **(8 Marks)**

|  |
| --- |
| 1. Based on the findings from Part A, where the optimal logistic regression model was identified, I will use this model as the foundation for incorporating polynomial terms, which can refine the model to capture more complex, non-linear patterns without starting from scratch.   From partA my best logistic regression model is as below  #model2=glm(KidneyDisease~BloodPressure+ElectricConductivity+pH+DissolvedOxygen+Turbidity+TotalDissolvedSolids, data = train,family =binomial)  # load the dataset  kidneydata=read.csv("KidneyData.csv")  attach(kidneydata)   1. Use str function to check data structures , remove patientid, KidneyDisease is int .It should be converted as a factor variable. And also other categorical variables converted into factor variables. Split the data 80% for training and 20% for testing with same setseed value and ratio with PartA.for model camparison.   #check data structures  head(kidneydata)  str(kidneydata)  kidneydata$KidneyDisease = as.factor(kidneydata$KidneyDisease)  kidneydata$Gender =as.factor(kidneydata$Gender)  kidneydata$SmokingStatus=as.factor(kidneydata$SmokingStatus)  #remove patientid  kidneydata=kidneydata[,-1]  set.seed(2)  tr1 = sample(1:nrow(kidneydata), round(nrow(kidneydata)\*0.8,0))  train = kidneydata[tr1, ] #defining training dataset  test=kidneydata[-tr1,]   1. A 10-fold cross-validation technique is chosen for its computational efficiency, requiring only 10 model fits, and its reduced tendency to overfit compared to Leave-One-Out Cross-Validation (LOOCV). 2. Perform 10-fold cross-validation on the polynomial models and record the MSE for each degree.   set.seed(1) #make it reproducible  poly\_order = c(1:10)  cv\_error\_10= rep (0,10)  for (i in 1:10) {  model\_10 = glm(KidneyDisease ~ poly(BloodPressure, i) +  poly(ElectricConductivity, i) +  poly(pH, i) +  poly(DissolvedOxygen, i) +  poly(Turbidity, i) +  poly(TotalDissolvedSolids, i),  data =train, family = binomial)  # Perform 10-fold cross-validation and store the error  cv\_error\_10[i] = cv.glm(kidneydata, model\_10, K = 10)$delta[1]  }  cv\_error\_10   1. Identify the lowest cross-validation error by plotting, which occurs with the third-degree polynomial. Fit a third-degree polynomial logistic regression model using the training dataset.   plot(poly\_order, cv\_error\_10, type = "b", xlab = "Order of polynomials", ylab = "10-Fold CV - MSE")    set.seed(2)  tr1 = sample(1:nrow(kidneydata), round(nrow(kidneydata)\*0.8,0))  train = kidneydata[tr1, ] #defining training dataset  test=kidneydata[-tr1,]   1. Summarize the model and check the p-values, since I already stated hypothesis testing in part A.Select the polynomial terms based on the significance codes. Then, choose the polynomial terms with p-values smaller than 0.05 to build the optimal model.   model\_1=glm(KidneyDisease~poly(BloodPressure,3)+  poly(ElectricConductivity,3)+  poly(pH,3)+  poly(DissolvedOxygen,3)+  poly(Turbidity,3)+  poly(TotalDissolvedSolids,3),  data = train,family = binomial  )  summary(model\_1)    ## Make predictions on the test set  predict=predict(model\_1,newdata = test,type = "response")  predict\_class=ifelse(predict>0.5,1,0)  ##calculate the misclassification rate  classification\_table=table(predict\_class,test$KidneyDisease)  misclassificationrate1=(classification\_table[1,2]+classification\_table[2,1])/sum(classification\_table)  ***misclassificationrate1is 0.13***  Fit Model 2 with the specified all significant predictors based on the significance codes  model\_2 = glm(KidneyDisease ~  poly(ElectricConductivity, 3)[, 1] +  poly(BloodPressure, 3)[,2] +  poly(pH, 3)[,1]+  poly(DissolvedOxygen, 3)[, 1] +  poly(TotalDissolvedSolids, 3)[,1]+  poly(Turbidity, 3)[, 1] +  poly(Turbidity, 3)[, 3],  data = train, family = binomial)  summary(model\_2)  ## Make predictions on the test set  predict=predict(model\_2,newdata = test,type = "response")  predict\_class=ifelse(predict>0.5,1,0)  ##calculate the misclassification rate  classification\_table=table(predict\_class,test$KidneyDisease)  misclassificationrate2=(classification\_table[1,2]+classification\_table[2,1])/sum(classification\_table)  ***misclassificationrate2 is 0.16***  Fit Model 3 with the specified significant predictors with p value less than 0.05  From model1  model\_3 = glm(KidneyDisease ~  poly(ElectricConductivity, 3)[, 1] +  poly(DissolvedOxygen, 3)[, 1] +  poly(Turbidity, 3)[, 1] +  poly(Turbidity, 3)[, 3],  data = train, family = binomial)  # Display the summary of Model 3  summary(model\_3)  ## Make predictions on the test set  predict=predict(model\_3,newdata = test,type = "response")  predict\_class=ifelse(predict>0.5,1,0)  ##calculate the misclassification rate  classification\_table=table(predict\_class,test$KidneyDisease)  misclassificationrate3=(classification\_table[1,2]+classification\_table[2,1])/sum(classification\_table)  ***misclassificationrate3 is 0.11***   1. Based on the lowest misclassification rate 0.11, choose the model where the polynomial terms have p-values less than 0.05. |

1. Give the resultant accepted model (i.e. write the model equation) based on your findings above. Justify your answer clearly. **(3 Marks)**

|  |
| --- |
| This is the resultant accepted model Based on the lowest misclassification rate 0.11 with comparison to other polynomial model.And there are three variables significantly associated with KidneyDisease for predicting the KidneyDisease .  model\_3 <- glm(KidneyDisease ~  poly(ElectricConductivity, 3)[, 1] +  poly(DissolvedOxygen, 3)[, 1] +  poly(Turbidity, 3)[, 1] +  poly(Turbidity, 3)[, 3],  data = train, family = binomial)    Using the coefficient above ,my model equation is as below:  logit(E(KidneyDisease))=2.449−45.783⋅poly(ElectricConductivity,3)[, 1]−9.039⋅poly(DissolvedOxygen,3) [,1]+10.910⋅poly(Turbidity,3)[,1]−12.139⋅poly(Turbidity,3) [,3] |

1. Use decision tree model to answer the research question. Clearly outline and explain each step of the process involved [*Hint: model building, improvement and evaluation*]. **(12 Marks)**

|  |
| --- |
| 1. Since the tree function requires categorical variables to be converted into factors, we first check the data structure, even though we have already converted categorical variables to factors in our code.import decision tree library and check the data structor.   library(tree)  str(train)   1. Build a model using the tree library with all variables in the train dataset. The reason for including all variables instead of just the significant ones found in part A is that cross-validation and pruning will help refine the tree and identify the most important variables.   tree\_kidneydata\_train<-tree(KidneyDisease~.,train)  plot(tree\_kidneydata\_train)  text(tree\_kidneydata\_train, pretty=0)  summary(tree\_kidneydata\_train)   1. Perform cross-validation to determine the optimal tree size. Based on the plot, the best size for the tree is 4.   par(mfrow=c(1,1))  set.seed(1)  cv\_kidneydata=cv.tree(tree\_kidneydata\_train, FUN = prune.misclass)  names(cv\_kidneydata)  plot(cv\_kidneydata$size, cv\_kidneydata$dev, type = "b")   1. Build the final tree model using the best size .   cv\_kidneydata=cv.tree(tree\_kidneydata\_train, FUN = prune.misclass)  pruned\_kidneydata=prune.misclass(tree\_kidneydata\_train, best = 4)  pruned\_kidneydata  summary(pruned\_kidneydata)  plot(pruned\_kidneydata)  text(pruned\_kidneydata, pretty=0)  summary(pruned\_kidneydata)   1. Evaluate its accuracy using the misclassification technique.The misclassification rate is calculated to be 0.06.   tree\_predicted<-predict(pruned\_kidneydata, test, type="class")  tab=table(tree\_predicted,test$KidneyDisease)  misrate=((tab[1,2]+tab[2,1])/sum(tab))  paste("Misclassification error rate is ",misrate) |

1. Give the resultant model and interpret it. Clearly describe the terminal nodes [i.e. list the profiles]. *[Include the relevant R output]* **(5 Marks)**

|  |
| --- |
| * This is my resultant model . * pruned\_kidneydata=prune.misclass(tree\_kidneydata\_train, best = 4)      * The three key variables (ElectricConductivity, Turbidity, and BloodPressure) played significant roles in determining the likelihood of kidney disease. * The model has 4 terminal nodes * The residual mean deviance is **0.437**, calculated as 173.1 / 396, where 173.1 represents the total deviance and 396 is the degrees of freedom (number of observations minus the number of terminal nodes)     Given my research question in Part A—“Which factors are significantly associated with the presence of kidney disease among patients in the suburbs, and how can we predict the likelihood of kidney disease?” The pruned tree model is fairly accurate with a misclassification error rate of 0.06, which is by far the lowest model . ElectricConductivity is the most important predictor in determining the kidneydiseas.next one is tubitity , and the last one is blood pressure.    describe the terminal nodes    Getting KidneyDisease(1) profile conditions:  when electricconductivity < 320.95  when electricconductivity >320.95 and Turbidity >1 and bloodpressure>140  Not getting KidneyDisease(0) profile conditions  when electricconductivity > 320.95 and Turbidity <1  when electricconductivity > 320.95 and Turbidity >1 and bloodpressure < 140 |

1. Compare the different resultant models (Part A Question 5, Part B Question 2 and Question 4) you obtained above. **(12 Marks)**

|  |
| --- |
| Logistic Regression  glm\_model=glm(KidneyDisease~BloodPressure+ElectricConductivity+pH+DissolvedOxygen+Turbidity+TotalDissolvedSolids, data = train,family =binomial)  Residual Deviance: 211.21  AIC: 225.21  Misclassification rate is 0.12  There are 6 variables significantly related  Polynomial Logistic Regression  Poly\_model = glm(KidneyDisease ~ poly(ElectricConductivity, 3)[, 1] + poly(DissolvedOxygen, 3)[, 1] + poly(Turbidity, 3)[, 1] + poly(Turbidity, 3)[, 3], data = train, family =binomial)  Residual Deviance: 241.55  AIC: 251.55  Misclassificationrate error rate is 0.11  There 3 factors significantly related.    Decision tree Model  pruned\_kidneydata=prune.misclass(tree\_kidneydata\_train, best = 4)  Variables actually used in tree construction:  [1] "ElectricConductivity" "Turbidity" "BloodPressure"  Residual mean deviance: 0.437 = 173.1 (the lowest, indicating the best fit) / 396  Misclassification error rate is 0.06 (the lowest, most accurate classification)  There are 3 factors significantly related  Logistic Regression:  Residual Deviance: 211.21, which is lower than the Polynomial Logistic Regression model but higher than the Decision Tree model.  AIC: 225.21, indicating the model's goodness of fit; it is lower than that of the Polynomial Logistic Regression, suggesting it fits the data better.  Misclassification Rate: 0.12, slightly higher than the Polynomial Logistic Regression model but lower than that of the Decision Tree model.  Variables: 6 variables are significantly related.  Summary: This model provides a reasonable fit but has a higher misclassification rate compared to both the Polynomial Logistic Regression and Decision Tree models. It utilizes more variables and provides detailed p-values and odds ratios for all 6 variables (Blood Pressure, Electric Conductivity, pH, Dissolved Oxygen, Turbidity, Total Dissolved Solids), showing which factors are significantly associated with kidney disease.  Polynomial Logistic Regression:  Residual Deviance: 241.55, the highest among all models, indicating a poorer fit.  AIC: 251.55, also the highest, showing the worst overall fit.  Misclassification Rate: 0.11, slightly lower than that of the Logistic Regression model but still higher than the Decision Tree.  Variables: 3 variables are significantly related.  Summary: This model performs slightly better in terms of misclassification than the Logistic Regression model but has the worst residual deviance and AIC. It simplifies the model by using fewer variables but is the least well-fitting model overall.  Decision Tree Model:  Residual Deviance: 173.1, the lowest, suggesting the best fit.  Misclassification Rate: 0.06, the lowest, indicating the most accurate classification.  Variables: 3 variables are significantly related.  Summary: The Decision Tree model is the best performer, offering the lowest misclassification rate and the best fit with the lowest residual deviance. It achieves this with only 3 variables, making it the most efficient model. While it provides clarity on the most influential factors, it doesn’t offer detailed information on how much each factor increases or decreases the likelihood of kidney disease. |

1. Give the final accepted model based on your findings above and Part A. Justify your answer. **(5 Marks)**

|  |
| --- |
| Final Accepted Model: Logistic Regression Model  Given my research question in Part A—“Which factors are significantly associated with the presence of kidney disease among patients in the suburbs, and how can we predict the likelihood of kidney disease?”—the final accepted model should not only prioritize accuracy but also provide insights into the significant factors associated with kidney disease. Although the decision tree has a lower misclassification rate (0.06), the logistic regression  identifies the significant factors associated with the disease. While the decision tree has a lower misclassification rate, the logistic regression model offers better interpretability in terms of statistical significance and the ability to predict probabilities, both of which are essential for answering the research question.  Justification:  Fore example the logistic regression model provides clear information about the statistical significance of predictors, helping to identify which variables are most associated with the presence of kidney disease. The p-values from the model show that:  Blood Pressure (p = 0.0003),Electric Conductivity (p < 9.29e-09),pH (p = 0.0078),Dissolved,Oxygen (p = 0.0079),Turbidity (p = 7.3e-05),Total Dissolved Solids (p = 1.03e-06)  These are all significant predictors of kidney disease. This aligns with the research question's aim to determine which factors are associated with the presence of kidney disease.  Predicting the Likelihood of Kidney Disease:  While the decision tree is accurate, it does not provide insights into likelihood in the same probabilistic manner as logistic regression.  Logistic regression provides an interpretation of coefficients that quantify the effect of each predictor on the likelihood of kidney disease. For example, higher blood pressure and turbidity increase the likelihood, while higher electric conductivity and pH reduce the likelihood. This is valuable for clinical interpretation and aligns with the research question’s focus on identifying risk factors. |

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. **(10 Marks)**

|  |
| --- |
| 1. Perform Principal Component Analysis (PCA) for the kidney dataset. The reason for this is that the kidney dataset has 18 variables excluding the patient ID. We need to reduce the dimensionality tocapture the essence of the data. PCA is one of the dimension reduction techniques. 2. Since Principal Component Analysis is applicable to numeric variables, I will only select the numeric variables.   str(kidneydata)  numerickidney\_data <- kidneydata %>%select\_if(is.numeric)  str(numerickidney\_data)   1. Perform Principal Component Analysis on the complete numeric kidney\_data dataset, setting scale to TRUE to facilitate the comparison of different variables.   pr.out=prcomp(numerickidney\_data,scale. = T)   1. Check the summary of the PCA results using summary(pr.out) to see the proportion of variance. PC1 and PC2 account for the highest proportions of variance, while PC 1 ~ 12 t account for almost 90% of the total variance. Additionally, create a plot to visualize this.   pr.var = pr.out$sdev^2  pve = pr.var/sum(pr.var)  #plot the Proportion of Variance Explained and Cumulative Proportion of Variance Explained  plot(pve,xlab="Principal Component",ylab="Proportion of Variance Explained",  ylim = c(0,1), type = 'b')  plot(cumsum(pve),xlab="Principal Component",ylab="Cumulative Proportion of Variance Explained",  ylim = c(0,1), type = 'b')       1. Check the standard deviations of the original variables.   High Variability Variables:  Blood Sugar, Electric Conductivity, Total Dissolved Solids, Cholesterol, Blood Pressure, Age, BMI.  The variables with high variability should be prioritized for intervention and monitoring, as they may have a more direct impact.  Low Variability Variables:  Dissolved Oxygen, pH, Nitrate, Nitrite, Lead, Arsenic.   1. Check the loadings for PC1 and PC2. Draw the biplot and interpret it. Given that there are 500 observations and many variables, I encountered overlapping issues that made it difficult to see the relationships in the biplot. To address this, I used the factoextra library and the fviz\_pca\_var function to fix it   #check the loading of pc1 and pc2  pr.out$rotation[,1:2]  pr.out  #Draw the biplot and interpret it  biplot(pr.out, scale = 0)  #to avoid overlapping  library(factoextra)  fviz\_pca\_var(pr.out, col.var = "contrib",  gradient.cols = c("blue", "red"),  repel = TRUE, # Avoid text overlapping  title = "PCA - Variable Contribution")      PC1 may indicate Environmental Factors  Electric Conductivity (0.672) and Total Dissolved Solids (0.681) are strongly positively related, Turbidity (0.151) is a weaker loading .indicate Connection Between Water Quality and Health, which means water quality issues that could lead to health problems if consumed over long periods. suggest that poor water quality may be a significant risk factor for kidney disease.  PC2 may indicate Medical Factors  BloodPressure (-0.659) and Age (-0.659): Both variables are strongly negatively correlated highlights that these two factors are likely the most critical health-related indicators affecting kidney disease, and bloodsuger has a weak negative loading but the highest variance . indicates that as age increases, the associated health factors worsen, high blood pressure and high bloodsuger can damage blood vessels in the kidneys over time. |

1. What are your conclusion and recommendations for this problem? [Hint: Use your results and findings from previous questions to answer this question] **(5 Marks)**

|  |
| --- |
| Conclusions   * Through PCA analysis, I identified age and blood pressure as critical factors negatively correlated with kidney health. Older adults and those with high blood pressure are at greater risk of developing kidney disease, highlighting the importance of monitoring and managing these conditions. * Additionally, there is an impact of water quality: Environmental factors, particularly electric conductivity and total dissolved solids, showed strong positive correlations with kidney disease risk. Poor water quality, characterized by these factors, is likely contributing to the rising incidence of kidney disease in the affected suburban areas.   Recommendations   * Enhanced Water Quality Monitoring: Implement regular testing of drinking water sources for electric conductivity, total dissolved solids, and other contaminants. This can help identify areas at risk and prioritize interventions,ensuring safe drinking water should be a key focus of public health efforts in the affected areas. * Encourage Older Adults to Be Aware of Blood Pressure and blood suger: Emphasize the significance of managing blood pressure and monitoring kidney health, especially among older adults. |

--- End of questions ---

APPENDIX

[Attach all your R codes and outputs here.]

reportB.R

xerzat

2024-10-06

# load the dataset  
kidneydata=read.csv("KidneyData.csv")  
attach(kidneydata)  
#check data structures  
head(kidneydata)

## PatientID Age Gender BloodPressure BloodSugar Cholesterol BMI SmokingStatus  
## 1 TIW5219 120 Female 118 155.8 165 31.7 Former  
## 2 QLJ3151 10 Female 143 162.5 214 23.9 Never  
## 3 GRL2542 58 Female 300 120.8 222 16.3 Former  
## 4 WMM4122 22 Female 20 154.2 212 21.9 Never  
## 5 LPP8404 52 Female 150 158.9 600 23.8 Current  
## 6 CIH1298 53 Female 141 131.6 199 18.3 Former  
## ElectricConductivity pH DissolvedOxygen Turbidity TotalDissolvedSolids  
## 1 336.2 7.40 9.57 1.44 455.4  
## 2 297.3 7.48 8.49 1.21 423.2  
## 3 377.9 7.49 8.18 0.88 434.3  
## 4 312.0 6.03 7.35 1.15 400.4  
## 5 222.4 6.77 7.40 0.73 349.4  
## 6 422.0 7.34 8.00 0.71 426.6  
## NitriteLevel NitrateLevel LeadConcentration ArsenicConcentration Humidity  
## 1 0.165 1.97 0.0099 0.0063 48.7  
## 2 0.075 1.74 0.0120 0.0062 65.3  
## 3 0.005 1.40 0.0173 0.0092 93.2  
## 4 0.088 0.88 0.0133 0.0086 67.4  
## 5 0.119 0.71 0.0155 0.0011 43.3  
## 6 0.076 1.00 0.0050 0.0090 57.6  
## KidneyDisease  
## 1 0  
## 2 1  
## 3 0  
## 4 1  
## 5 1  
## 6 0

str(kidneydata)

## '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 0.0128 0.0145 ...  
## $ ArsenicConcentration: num 0.0063 0.0062 0.0092 0.0086 0.0011 0.009 0.0035 0.0062 0.0081 0.0046 ...  
## $ 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 ...

#There are 500 observations and 19 variables in the data set.KidneyDisease is int .It should be converted as a factor variable.  
#And other categorical variables converted into factor variables.  
kidneydata$KidneyDisease = as.factor(kidneydata$KidneyDisease)  
kidneydata$Gender =as.factor(kidneydata$Gender)  
kidneydata$SmokingStatus=as.factor(kidneydata$SmokingStatus)  
  
#remove patientid  
kidneydata=kidneydata[,-1]  
  
  
# from partA my best logistic regression model is as below  
#model2=glm(KidneyDisease~BloodPressure+ElectricConductivity+pH+DissolvedOxygen+Turbidity+TotalDissolvedSolids, data = train,family =binomial)  
#I choose 10-Fold Cross-Validation because it requires only 10 model fits, making it computationally less expensive than LOOCV  
library(boot)  
  
#split the data: 80% for training and 20% for testing with same setseed value and ratio with PartA.  
set.seed(2)  
tr1 = sample(1:nrow(kidneydata), round(nrow(kidneydata)\*0.8,0))  
train = kidneydata[tr1, ] #defining training dataset  
test=kidneydata[-tr1,]  
  
set.seed(2)  
poly\_order = c(1:10)  
cv\_error\_10= rep (0,10)  
for (i in 1:10) {  
model\_10 = glm(KidneyDisease ~ poly(BloodPressure, i) +   
 poly(ElectricConductivity, i) +   
 poly(pH, i) +   
 poly(DissolvedOxygen, i) +   
 poly(Turbidity, i) +   
 poly(TotalDissolvedSolids, i),   
 data = train, family = binomial)  
   
 # Perform 10-fold cross-validation and store the error  
 cv\_error\_10[i] = cv.glm(train, model\_10, K = 10)$delta[1]  
}

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

cv\_error\_10

## [1] 0.08756614 0.08864296 0.08573451 0.10388761 0.10811442 0.12096893  
## [7] 0.16500000 0.14250000 0.15250000 0.16750000

plot(poly\_order, cv\_error\_10, type = "b", xlab = "Order of polynomials", ylab = "10-Fold CV - MSE")

A graph with lines and numbers

Description automatically generated

#The lowest cross-validation error is at the third degree (cv\_error\_10[3]),   
#which indicates that the third-degree polynomial provides the best fit with minimal error.  
  
#check both train data and test just in case  
dim(train)

## [1] 400 18

dim(test)

## [1] 100 18

#Fit a polynomial logistic regression model on train data set  
model\_1=glm(KidneyDisease~poly(BloodPressure,3)+  
 poly(ElectricConductivity,3)+  
 poly(pH,3)+  
 poly(DissolvedOxygen,3)+  
 poly(Turbidity,3)+  
 poly(TotalDissolvedSolids,3),  
 data = train,family = binomial  
 )

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(model\_1)

##   
## Call:  
## glm(formula = KidneyDisease ~ poly(BloodPressure, 3) + poly(ElectricConductivity,   
## 3) + poly(pH, 3) + poly(DissolvedOxygen, 3) + poly(Turbidity,   
## 3) + poly(TotalDissolvedSolids, 3), family = binomial, data = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 8.160 2.517 3.242 0.001186 \*\*   
## poly(BloodPressure, 3)1 -17.237 28.020 -0.615 0.538432   
## poly(BloodPressure, 3)2 -48.766 25.211 -1.934 0.053068 .   
## poly(BloodPressure, 3)3 -61.896 49.043 -1.262 0.206917   
## poly(ElectricConductivity, 3)1 -143.156 66.337 -2.158 0.030926 \*   
## poly(ElectricConductivity, 3)2 81.937 51.605 1.588 0.112336   
## poly(ElectricConductivity, 3)3 -26.295 23.345 -1.126 0.260014   
## poly(pH, 3)1 -8.043 4.129 -1.948 0.051404 .   
## poly(pH, 3)2 2.170 4.136 0.524 0.599942   
## poly(pH, 3)3 5.272 4.147 1.271 0.203645   
## poly(DissolvedOxygen, 3)1 -12.827 4.532 -2.831 0.004646 \*\*   
## poly(DissolvedOxygen, 3)2 -5.102 4.666 -1.093 0.274222   
## poly(DissolvedOxygen, 3)3 -5.274 4.701 -1.122 0.261920   
## poly(Turbidity, 3)1 16.648 4.563 3.648 0.000264 \*\*\*  
## poly(Turbidity, 3)2 -1.143 4.507 -0.254 0.799769   
## poly(Turbidity, 3)3 -11.788 4.511 -2.613 0.008976 \*\*   
## poly(TotalDissolvedSolids, 3)1 -86.040 44.287 -1.943 0.052044 .   
## poly(TotalDissolvedSolids, 3)2 51.569 37.527 1.374 0.169386   
## poly(TotalDissolvedSolids, 3)3 -27.287 26.773 -1.019 0.308113   
## ---  
## 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: 168.46 on 381 degrees of freedom  
## AIC: 206.46  
##   
## Number of Fisher Scoring iterations: 11

## Make predictions on the test set  
predict=predict(model\_1,newdata = test,type = "response")  
predict\_class=ifelse(predict>0.5,1,0)  
  
##calculate the misclassification rate  
classification\_table=table(predict\_class,test$KidneyDisease)  
misclassificationrate1=(classification\_table[1,2]+classification\_table[2,1])/sum(classification\_table)  
misclassificationrate1

## [1] 0.13

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Fit Model 2 with the specified all significant predictors  
model\_2 = glm(KidneyDisease ~   
 poly(ElectricConductivity, 3)[, 1] +   
 poly(BloodPressure, 3)[,2] +  
 poly(pH, 3)[,1]+  
 poly(DissolvedOxygen, 3)[, 1] +   
 poly(TotalDissolvedSolids, 3)[,1]+  
 poly(Turbidity, 3)[, 1] +   
 poly(Turbidity, 3)[, 3],  
 data = train, family = binomial)  
summary(model\_2)

##   
## Call:  
## glm(formula = KidneyDisease ~ poly(ElectricConductivity, 3)[,   
## 1] + poly(BloodPressure, 3)[, 2] + poly(pH, 3)[, 1] + poly(DissolvedOxygen,   
## 3)[, 1] + poly(TotalDissolvedSolids, 3)[, 1] + poly(Turbidity,   
## 3)[, 1] + poly(Turbidity, 3)[, 3], family = binomial, data = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.7946 0.2954 9.461 < 2e-16 \*\*\*  
## poly(ElectricConductivity, 3)[, 1] -34.5928 5.9296 -5.834 5.41e-09 \*\*\*  
## poly(BloodPressure, 3)[, 2] -6.5862 3.4326 -1.919 0.055018 .   
## poly(pH, 3)[, 1] -6.9223 3.5702 -1.939 0.052516 .   
## poly(DissolvedOxygen, 3)[, 1] -9.9720 3.8439 -2.594 0.009481 \*\*   
## poly(TotalDissolvedSolids, 3)[, 1] -24.6792 5.4547 -4.524 6.06e-06 \*\*\*  
## poly(Turbidity, 3)[, 1] 13.7621 3.5849 3.839 0.000124 \*\*\*  
## poly(Turbidity, 3)[, 3] -11.2837 3.4297 -3.290 0.001002 \*\*   
## ---  
## 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: 212.35 on 392 degrees of freedom  
## AIC: 228.35  
##   
## Number of Fisher Scoring iterations: 6

## Make predictions on the test set  
predict=predict(model\_2,newdata = test,type = "response")  
predict\_class=ifelse(predict>0.5,1,0)  
  
##calculate the misclassification rate  
classification\_table=table(predict\_class,test$KidneyDisease)  
misclassificationrate2=(classification\_table[1,2]+classification\_table[2,1])/sum(classification\_table)  
misclassificationrate2

## [1] 0.16

# Fit Model 3 with the specified significant predictors with p value less than 0.05 from model 1  
model\_3 = glm(KidneyDisease ~   
 poly(ElectricConductivity, 3)[, 1] +   
 poly(DissolvedOxygen, 3)[, 1] +   
 poly(Turbidity, 3)[, 1] +   
 poly(Turbidity, 3)[, 3],   
 data = train, family = binomial)  
  
# Display the summary of Model 3  
summary(model\_3)

##   
## Call:  
## glm(formula = KidneyDisease ~ poly(ElectricConductivity, 3)[,   
## 1] + poly(DissolvedOxygen, 3)[, 1] + poly(Turbidity, 3)[,   
## 1] + poly(Turbidity, 3)[, 3], family = binomial, data = train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 2.449 0.246 9.956 < 2e-16 \*\*\*  
## poly(ElectricConductivity, 3)[, 1] -45.783 5.263 -8.698 < 2e-16 \*\*\*  
## poly(DissolvedOxygen, 3)[, 1] -9.039 3.462 -2.611 0.009038 \*\*   
## poly(Turbidity, 3)[, 1] 10.910 3.340 3.267 0.001089 \*\*   
## poly(Turbidity, 3)[, 3] -12.139 3.413 -3.557 0.000375 \*\*\*  
## ---  
## 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: 241.55 on 395 degrees of freedom  
## AIC: 251.55  
##   
## Number of Fisher Scoring iterations: 6

## Make predictions on the test set  
predict=predict(model\_3,newdata = test,type = "response")  
predict\_class=ifelse(predict>0.5,1,0)  
  
##calculate the misclassification rate  
classification\_table=table(predict\_class,test$KidneyDisease)  
misclassificationrate3=(classification\_table[1,2]+classification\_table[2,1])/sum(classification\_table)  
misclassificationrate3

## [1] 0.11

##equation   
#logit(E(KidneyDisease))=2.449−45.783⋅poly(ElectricConductivity,3)[, 1]−9.039⋅poly(DissolvedOxygen,3) [,1]+10.910⋅poly(Turbidity,3)[,1]−12.139⋅poly(Turbidity,3) [,3]  
  
  
  
#decision tree library and check the data structor  
library(tree)  
str(train)

## 'data.frame': 400 obs. of 18 variables:  
## $ Age : int 71 71 61 83 87 42 50 52 46 77 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 2 1 1 1 2 1 2 1 1 1 ...  
## $ BloodPressure : int 175 172 147 170 145 144 157 161 155 172 ...  
## $ BloodSugar : num 62.8 86.5 89 110.8 39.1 ...  
## $ Cholesterol : int 243 227 196 253 200 179 204 215 204 240 ...  
## $ BMI : num 14.7 22.4 31 15.7 32.2 23.4 19.8 19.2 16.2 22.1 ...  
## $ SmokingStatus : Factor w/ 3 levels "Current","Former",..: 3 1 3 3 1 3 3 1 2 3 ...  
## $ ElectricConductivity: num 232 246 365 290 256 ...  
## $ pH : num 6.76 7.06 6.95 7.54 7.51 7.92 6.39 6.56 6.72 6.16 ...  
## $ DissolvedOxygen : num 8.36 8.29 8.85 8.96 8.25 7.79 6.37 9.11 8.06 8.27 ...  
## $ Turbidity : num 1.01 0.94 0.67 0.77 0.78 0.72 1.28 1.18 1.26 0.89 ...  
## $ TotalDissolvedSolids: num 391 422 469 411 355 ...  
## $ NitriteLevel : num 0.078 0.043 0.08 0.052 0.1 0.076 0.076 0.055 0.019 0.119 ...  
## $ NitrateLevel : num 0.48 0.77 0.33 0.67 1.41 0.73 1.94 0.7 1.16 0.28 ...  
## $ LeadConcentration : num 0.0189 0.0081 0.011 0.0123 0.0105 0.0146 0.0093 0.0056 0.0091 0.0116 ...  
## $ ArsenicConcentration: num 0.0061 0.0015 0.0075 0.0057 0.0067 0.0036 0.0063 0.0053 0.0038 0.0072 ...  
## $ Humidity : num 70.2 56.7 67.4 57.9 52.2 66.4 61.4 42.5 48.7 45.1 ...  
## $ KidneyDisease : Factor w/ 2 levels "0","1": 2 2 1 2 2 2 2 2 2 2 ...

#The tree() function expects categorical variables like the Gender and SmokingStatus to be factors,I already converted them from the beginning  
tree\_kidneydata\_train<-tree(KidneyDisease~.,train)  
plot(tree\_kidneydata\_train)  
text(tree\_kidneydata\_train, pretty=0)

A diagram of a number of objects

Description automatically generated with medium confidence

summary(tree\_kidneydata\_train)

##   
## Classification tree:  
## tree(formula = KidneyDisease ~ ., data = train)  
## Variables actually used in tree construction:  
## [1] "ElectricConductivity" "TotalDissolvedSolids" "pH"   
## [4] "BloodSugar" "NitriteLevel" "Turbidity"   
## [7] "Cholesterol" "BMI" "DissolvedOxygen"   
## [10] "BloodPressure"   
## Number of terminal nodes: 15   
## Residual mean deviance: 0.1028 = 39.56 / 385   
## Misclassification error rate: 0.0275 = 11 / 400

#cross validation  
par(mfrow=c(1,1))  
set.seed(1)  
cv\_kidneydata=cv.tree(tree\_kidneydata\_train, FUN = prune.misclass)  
names(cv\_kidneydata)

## [1] "size" "dev" "k" "method"

plot(cv\_kidneydata$size, cv\_kidneydata$dev, type = "b")

A graph with lines and dots

Description automatically generated

# Select the best size of the tree model  
#According the plot , best size of the tree model is 4.  
cv\_kidneydata=cv.tree(tree\_kidneydata\_train, FUN = prune.misclass)  
pruned\_kidneydata=prune.misclass(tree\_kidneydata\_train, best = 4)  
pruned\_kidneydata

## node), split, n, deviance, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 400 408.50 1 ( 0.20750 0.79250 )   
## 2) ElectricConductivity < 320.95 269 98.09 1 ( 0.04461 0.95539 ) \*  
## 3) ElectricConductivity > 320.95 131 180.70 0 ( 0.54198 0.45802 )   
## 6) Turbidity < 1 64 61.77 0 ( 0.81250 0.18750 ) \*  
## 7) Turbidity > 1 67 79.90 1 ( 0.28358 0.71642 )   
## 14) BloodPressure < 140 21 13.21 0 ( 0.90476 0.09524 ) \*  
## 15) BloodPressure > 140 46 0.00 1 ( 0.00000 1.00000 ) \*

summary(pruned\_kidneydata)

##   
## Classification tree:  
## snip.tree(tree = tree\_kidneydata\_train, nodes = c(14L, 2L, 6L  
## ))  
## Variables actually used in tree construction:  
## [1] "ElectricConductivity" "Turbidity" "BloodPressure"   
## Number of terminal nodes: 4   
## Residual mean deviance: 0.437 = 173.1 / 396   
## Misclassification error rate: 0.065 = 26 / 400

plot(pruned\_kidneydata)  
text(pruned\_kidneydata, pretty=0)

A diagram of a flowchart

Description automatically generated

summary(pruned\_kidneydata)

##   
## Classification tree:  
## snip.tree(tree = tree\_kidneydata\_train, nodes = c(14L, 2L, 6L  
## ))  
## Variables actually used in tree construction:  
## [1] "ElectricConductivity" "Turbidity" "BloodPressure"   
## Number of terminal nodes: 4   
## Residual mean deviance: 0.437 = 173.1 / 396   
## Misclassification error rate: 0.065 = 26 / 400

#The three key variables (ElectricConductivity, Turbidity, and BloodPressure)   
#played significant roles in determining the likelihood of kidney disease.  
  
  
#check the model accuracy  
tree\_predicted<-predict(pruned\_kidneydata, test, type="class")  
tab=table(tree\_predicted,test$KidneyDisease)  
misrate=((tab[1,2]+tab[2,1])/sum(tab))  
paste("Misclassification error rate is ",misrate)

## [1] "Misclassification error rate is 0.06"

#9. 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  
options(scipen = 999)  
str(kidneydata)

## 'data.frame': 500 obs. of 18 variables:  
## $ Age : int 120 10 58 22 52 53 76 45 57 30 ...  
## $ Gender : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 2 2 1 ...  
## $ 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 : Factor w/ 3 levels "Current","Former",..: 2 3 2 3 1 2 1 1 2 2 ...  
## $ 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 0.0128 0.0145 ...  
## $ ArsenicConcentration: num 0.0063 0.0062 0.0092 0.0086 0.0011 0.009 0.0035 0.0062 0.0081 0.0046 ...  
## $ Humidity : num 48.7 65.3 93.2 67.4 43.3 57.6 50.8 70.5 55.6 72.9 ...  
## $ KidneyDisease : Factor w/ 2 levels "0","1": 1 2 1 2 2 1 2 2 1 2 ...

#since we use only numerical variables for Principal Component Analysis and unsupervised learning is for dataset that has no target variaable ,   
#I will select only numerical variables,so that there is no factor and categorical variables   
numerickidney\_data <- kidneydata %>%select\_if(is.numeric)  
str(numerickidney\_data)

## 'data.frame': 500 obs. of 15 variables:  
## $ Age : int 120 10 58 22 52 53 76 45 57 30 ...  
## $ 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 ...  
## $ 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 0.0128 0.0145 ...  
## $ ArsenicConcentration: num 0.0063 0.0062 0.0092 0.0086 0.0011 0.009 0.0035 0.0062 0.0081 0.0046 ...  
## $ Humidity : num 48.7 65.3 93.2 67.4 43.3 57.6 50.8 70.5 55.6 72.9 ...

#Perform Principal Component Analysis for the numeric kidney\_data dataset.  
pr.out=prcomp(numerickidney\_data,scale. = T)  
  
#check the Cumulative Proportion of Variance Explained  
summary(pr.out)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 1.3294 1.2551 1.10540 1.07599 1.05867 1.03697 1.01698  
## Proportion of Variance 0.1178 0.1050 0.08146 0.07718 0.07472 0.07169 0.06895  
## Cumulative Proportion 0.1178 0.2228 0.30431 0.38149 0.45621 0.52790 0.59685  
## PC8 PC9 PC10 PC11 PC12 PC13 PC14  
## Standard deviation 0.99134 0.97911 0.9542 0.93496 0.89893 0.8780 0.68134  
## Proportion of Variance 0.06552 0.06391 0.0607 0.05828 0.05387 0.0514 0.03095  
## Cumulative Proportion 0.66236 0.72628 0.7870 0.84526 0.89913 0.9505 0.98147  
## PC15  
## Standard deviation 0.52719  
## Proportion of Variance 0.01853  
## Cumulative Proportion 1.00000

pr.var = pr.out$sdev^2  
pr.var

## [1] 1.7673658 1.5753231 1.2219131 1.1577623 1.1207757 1.0753101 1.0342530  
## [8] 0.9827591 0.9586643 0.9105472 0.8741552 0.8080768 0.7709463 0.4642180  
## [15] 0.2779300

pve = pr.var/sum(pr.var)  
pve

## [1] 0.11782438 0.10502154 0.08146087 0.07718416 0.07471838 0.07168734  
## [7] 0.06895020 0.06551727 0.06391095 0.06070314 0.05827702 0.05387178  
## [13] 0.05139642 0.03094787 0.01852867

#plot the Proportion of Variance Explained and Cumulative Proportion of Variance Explained  
plot(pve,xlab="Principal Component",ylab="Proportion of Variance Explained",  
 ylim = c(0,1), type = 'b')

A graph with lines and numbers

Description automatically generated

plot(cumsum(pve),xlab="Principal Component",ylab="Cumulative Proportion of Variance Explained",  
 ylim = c(0,1), type = 'b')

A graph with lines and numbers

Description automatically generated

#This gives the standard deviances of the original variables.  
sd\_rank=sort(pr.out$scale,decreasing = T)  
sd\_rank

## BloodSugar ElectricConductivity TotalDissolvedSolids   
## 53.553034938 51.921610754 36.907204860   
## Cholesterol BloodPressure Age   
## 34.076312802 20.837135322 20.390639823   
## Humidity BMI DissolvedOxygen   
## 9.324119485 5.241010183 0.943478818   
## pH NitrateLevel Turbidity   
## 0.516125639 0.492107997 0.208735998   
## NitriteLevel LeadConcentration ArsenicConcentration   
## 0.049100377 0.004748077 0.002050100

#check the loading of pc1 and pc2  
pr.out$rotation[,1:2]

## PC1 PC2  
## Age -0.0459113740 -0.65914163  
## BloodPressure -0.0171366753 -0.65935810  
## BloodSugar -0.0472074716 -0.01483059  
## Cholesterol -0.1171941168 0.16402495  
## BMI 0.1234482828 0.12544944  
## ElectricConductivity 0.6721372605 -0.01232257  
## pH 0.0185612764 -0.04239933  
## DissolvedOxygen -0.0008670074 -0.11470278  
## Turbidity 0.1514371007 0.01516362  
## TotalDissolvedSolids 0.6813096824 -0.01833640  
## NitriteLevel 0.0290639644 -0.07332969  
## NitrateLevel 0.0110255389 -0.12420629  
## LeadConcentration 0.0455052516 -0.05601159  
## ArsenicConcentration 0.1141595867 -0.20022036  
## Humidity -0.1053410225 -0.09063744

pr.out

## Standard deviations (1, .., p=15):  
## [1] 1.3294231 1.2551187 1.1054018 1.0759936 1.0586670 1.0369716 1.0169823  
## [8] 0.9913421 0.9791140 0.9542259 0.9349627 0.8989309 0.8780355 0.6813354  
## [15] 0.5271907  
##   
## Rotation (n x k) = (15 x 15):  
## PC1 PC2 PC3 PC4  
## Age -0.0459113740 -0.65914163 -0.00331068 0.0007868643  
## BloodPressure -0.0171366753 -0.65935810 0.12575360 0.0485900614  
## BloodSugar -0.0472074716 -0.01483059 0.06609886 -0.0694296011  
## Cholesterol -0.1171941168 0.16402495 0.54842454 0.0058012486  
## BMI 0.1234482828 0.12544944 -0.16375907 0.1394708012  
## ElectricConductivity 0.6721372605 -0.01232257 0.11882945 0.0696862757  
## pH 0.0185612764 -0.04239933 -0.25653457 0.1340141799  
## DissolvedOxygen -0.0008670074 -0.11470278 0.37861357 -0.4024110990  
## Turbidity 0.1514371007 0.01516362 -0.31759843 0.0297263305  
## TotalDissolvedSolids 0.6813096824 -0.01833640 0.12201057 -0.0167943359  
## NitriteLevel 0.0290639644 -0.07332969 -0.09836519 -0.5445937476  
## NitrateLevel 0.0110255389 -0.12420629 -0.34369900 -0.2319108370  
## LeadConcentration 0.0455052516 -0.05601159 0.38010472 0.0948584163  
## ArsenicConcentration 0.1141595867 -0.20022036 -0.17189449 0.1667177623  
## Humidity -0.1053410225 -0.09063744 0.10655270 0.6326593603  
## PC5 PC6 PC7 PC8  
## Age -0.06063086 0.01318603 0.13540469 -0.0854257437  
## BloodPressure -0.06178029 0.09424886 0.18564220 0.0281789214  
## BloodSugar -0.43202619 -0.34258296 -0.14458692 -0.3450597093  
## Cholesterol -0.19183490 -0.02657233 0.23627498 0.0570031776  
## BMI 0.55186933 0.02493300 0.15361969 -0.2571974713  
## ElectricConductivity -0.07204558 0.05062802 0.05646538 0.0002281068  
## pH 0.15076360 -0.49563457 0.52509279 0.1738473074  
## DissolvedOxygen 0.29874330 -0.37652093 -0.13591367 0.0730364776  
## Turbidity -0.44775257 -0.28859623 -0.27505470 0.0982799712  
## TotalDissolvedSolids -0.06390056 0.09973275 0.06950287 0.0159992495  
## NitriteLevel 0.10222107 0.30458731 -0.13800392 -0.4160096791  
## NitrateLevel 0.03660827 0.21269550 -0.19630527 0.6250732588  
## LeadConcentration 0.25356094 -0.15247221 -0.40990544 0.3415674985  
## ArsenicConcentration 0.25757748 -0.37972388 -0.40811682 -0.2638181988  
## Humidity 0.01838546 0.28910622 -0.27720138 -0.0890003494  
## PC9 PC10 PC11 PC12  
## Age -0.15220162 -0.099906882 -0.16901815 -0.096251003  
## BloodPressure 0.09574016 -0.009514777 -0.11854476 0.047990311  
## BloodSugar 0.66335616 0.245035003 -0.13378891 -0.047535587  
## Cholesterol -0.19756353 0.120773583 -0.25700372 0.649130620  
## BMI 0.18413149 0.012728194 -0.69738690 0.042207835  
## ElectricConductivity 0.05873296 0.101421972 0.14158416 0.014555128  
## pH 0.26518508 -0.187102541 0.32399615 0.310459355  
## DissolvedOxygen -0.06871756 0.262176196 -0.01413726 -0.148150374  
## Turbidity -0.25521823 -0.353377249 -0.39035981 0.147228787  
## TotalDissolvedSolids 0.01852713 0.024251640 -0.01142426 -0.004244713  
## NitriteLevel 0.15259316 -0.340238054 0.19139171 0.446295410  
## NitrateLevel 0.24529031 0.403248526 -0.13765252 0.282360660  
## LeadConcentration 0.28311262 -0.552157882 -0.04303052 0.004033538  
## ArsenicConcentration -0.32830167 0.287655319 0.17026601 0.314896540  
## Humidity 0.19831843 0.098227760 0.15142407 0.211419051  
## PC13 PC14 PC15  
## Age 0.09452981 -0.673751187 0.008732963  
## BloodPressure -0.02655673 0.685005441 -0.057194268  
## BloodSugar 0.13773036 -0.061849843 0.036646723  
## Cholesterol 0.11745389 -0.083407591 -0.013599849  
## BMI -0.03917506 0.003245223 -0.063884543  
## ElectricConductivity -0.01161139 -0.091927656 -0.689624150  
## pH -0.15355903 -0.076706589 0.060050735  
## DissolvedOxygen -0.57403511 -0.040097483 -0.002489630  
## Turbidity -0.36413721 0.078474674 -0.056366357  
## TotalDissolvedSolids -0.01061604 -0.006098606 0.707082680  
## NitriteLevel -0.12721007 -0.022584308 -0.025275086  
## NitrateLevel 0.05914423 -0.108403657 -0.002419462  
## LeadConcentration 0.28418462 -0.033756455 -0.016826807  
## ArsenicConcentration 0.33369347 0.101242858 0.050613693  
## Humidity -0.50825812 -0.142285744 0.072228577

#Draw the biplot and interpret it  
biplot(pr.out,scale = 0,cex=0.4)

A diagram of a graph

Description automatically generated with medium confidence