ECON 412 - Project 2 - Part 1

A. Riad, B. Graf, M. Caufield, Q. He, Y. Luo

05-26-2021

Contents

Team Members:

- Ashtin Riad (805656966)
- Benedikt Graf (105652212)
- Madelyn Caufield (505657057)
- Qiumeng He (305524290)
- Yansha Luo (505646846)

Introduction

This project is to fit several classification models, including linear and non-linear ones to a dataset about fetal health. We will fit 5 models to the data: logistic regression, LDA, QDA, KNN and K-means clustering. We will then use cross validation to evaluate each model's performance and select the preferred one.

Data Description

"This dataset contains 2126 records of features extracted from Cardiotocogram exams, which were then classified by three expert obstetritians into 3 classes: Normal, Suspect, Pathological" (Kaggle, 2021).

1 Dependent Variable:

- Three classes of fetal health (Normal-1, Suspect-2, Pathological-3)

21 Independent Variables

- Baseline Fetal Heart Rate (FHR): Baseline Fetal Heart Rate (FHR)
- accelerations: Number of accelerations per second
- fetal_movement: Number of fetal movements per second
- uterine_contractions: Number of uterine contractions per second
- light_decelerations: Number of LDs per second
- severe_decelerations: Number of SDs per second
- prolongued_decelerations: Number of PDs per second
- abnormal_short_term_variability: Percentage of time with abnormal short term variability
- mean_value_of_short_term_variability: Mean value of short term variability
- percentage_of_time_with_abnormal_long_term_variability:Percentage of time with abnormal long term variability
- mean_value_of_long_term_variability: Mean value of long term variability
- histogram_width: Width of the histogram made using all values from a record
- histogram_min: Histogram minimum value
- histogram_max: Histogram maximum value
- histogram_number_of_peaks: Number of peaks in the exam histogram
- histogram_number_of_zeroes: Number of zeroes in the exam histogram
- histogram_mode: Hist mode
- histogram_mean: Hist mean
- histogram_median: Hist median
- histogram_variance: Hist variance
- histogram_tendency: Histogram trend
- fetal_health:

Logistic regression

The fetal health is used as the dependent variable and all other variables have been used as predictors. The variable fetal health has 3 levels, and hence, multinomial logistic regression would be applied here. Below is the summary of the model fit.

The data has been divided into training and test set with 70-30% split.

```
myfile <- read.csv("fetal_health.csv", header = T)
myfile$fetal_health <- as.factor(myfile$fetal_health)
myfile$fetal_health <- relevel(myfile$fetal_health, ref = 1)

set.seed(100)
tr_samp <- sample(nrow(myfile), floor(0.7*nrow(myfile)))

training <- myfile[tr_samp,]
testing <- myfile[-tr_samp,]

model_logi <- multinom(fetal_health ~., training)
```

```
## # weights:  69 (44 variable)
## initial  value 1634.735086
## iter  10 value 692.043284
## iter  20 value 629.148946
## iter  30 value 446.735404
```

```
## iter  40 value 401.246918
## iter  50 value 397.554801
## iter  60 value 352.979807
## iter  70 value 343.911150
## iter  80 value 327.479129
## iter  90 value 323.573617
## iter  90 value 323.573615
## final  value 323.573615
## converged
```

```
stargazer(model_logi, type = "text")
```

```
##
## ===============================================================================
##                                                  Dependent variable:
##                                              ---------------------------
##                                                    2            3
##                                                   (1)          (2)
## -------------------------------------------------------------------------------
## baseline.value                                   0.007        0.547***
##                                                 (0.034)      (0.055)
##
## accelerations                                -1,034.242***   -86.558***
##                                                 (0.003)      (0.003)
##
## fetal_movement                                 11.198***     20.088***
##                                                 (0.249)      (0.277)
##
## uterine_contractions                          -245.926***   -477.289***
##                                                 (0.001)      (0.007)
##
## light_decelerations                            -75.787***   -125.901***
##                                                 (0.002)      (0.002)
##
## severe_decelerations                            -2.813***    -1.209***
##                                                (0.00005)    (0.00002)
##
## prolongued_decelerations                       144.650***    83.159***
##                                                (0.0001)      (0.0003)
##
## abnormal_short_term_variability                 0.069***      0.131***
##                                                 (0.011)      (0.022)
##
## mean_value_of_short_term_variability             -0.549*      -2.420***
##                                                 (0.301)      (0.555)
##
## percentage_of_time_with_abnormal_long_term_variability    0.014**      0.061***
##                                                 (0.007)      (0.012)
```

```
##
## mean_value_of_long_term_variability                    -0.047         -0.041
##                                                        (0.041)        (0.084)
##
## histogram_width                                         0.005         0.016**
##                                                        (0.005)        (0.007)
##
## histogram_min                                         0.025***        0.025**
##                                                        (0.008)        (0.013)
##
## histogram_max                                         0.030***        0.041***
##                                                        (0.011)        (0.013)
##
## histogram_number_of_peaks                             0.180***         -0.203
##                                                        (0.061)        (0.125)
##
## histogram_number_of_zeroes                             -0.292          0.454
##                                                        (0.180)        (0.408)
##
## histogram_mode                                        -0.064**        -0.073*
##                                                        (0.028)        (0.043)
##
## histogram_mean                                        0.264***       -0.134***
##                                                        (0.058)        (0.044)
##
## histogram_median                                      -0.156**       -0.335***
##                                                        (0.069)        (0.065)
##
## histogram_variance                                    0.058***        0.070***
##                                                        (0.008)        (0.012)
##
## histogram_tendency                                     0.679**         0.326
##                                                        (0.307)        (0.463)
##
## Constant                                              -18.296***     -19.398***
##                                                        (1.974)        (0.594)
##
## ------------------------------------------------------------------------------
## Akaike Inf. Crit.                                      731.147        731.147
## ==============================================================================
## Note:                                                *p<0.1; **p<0.05; ***p<0.01
```

Now, we can use the model to look at the predictions on the test data.

```
pred_logi <- predict(model_logi, testing)

confusionMatrix(testing$fetal_health, pred_logi)
```

```
## Confusion Matrix and Statistics
```

```
##
##           Reference
## Prediction   1   2   3
##          1 469  28   1
##          2  33  54   5
##          3   7   7  34
##
## Overall Statistics
##
##                Accuracy : 0.873
##                  95% CI : (0.8447, 0.8979)
##     No Information Rate : 0.7978
##     P-Value [Acc > NIR] : 4.036e-07
##
##                   Kappa : 0.6398
##
##  Mcnemar's Test P-Value : 0.1548
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9214  0.60674  0.85000
## Specificity            0.7752  0.93078  0.97659
## Pos Pred Value         0.9418  0.58696  0.70833
## Neg Pred Value         0.7143  0.93590  0.98983
## Prevalence             0.7978  0.13950  0.06270
## Detection Rate         0.7351  0.08464  0.05329
## Detection Prevalence   0.7806  0.14420  0.07524
## Balanced Accuracy      0.8483  0.76876  0.91329
```

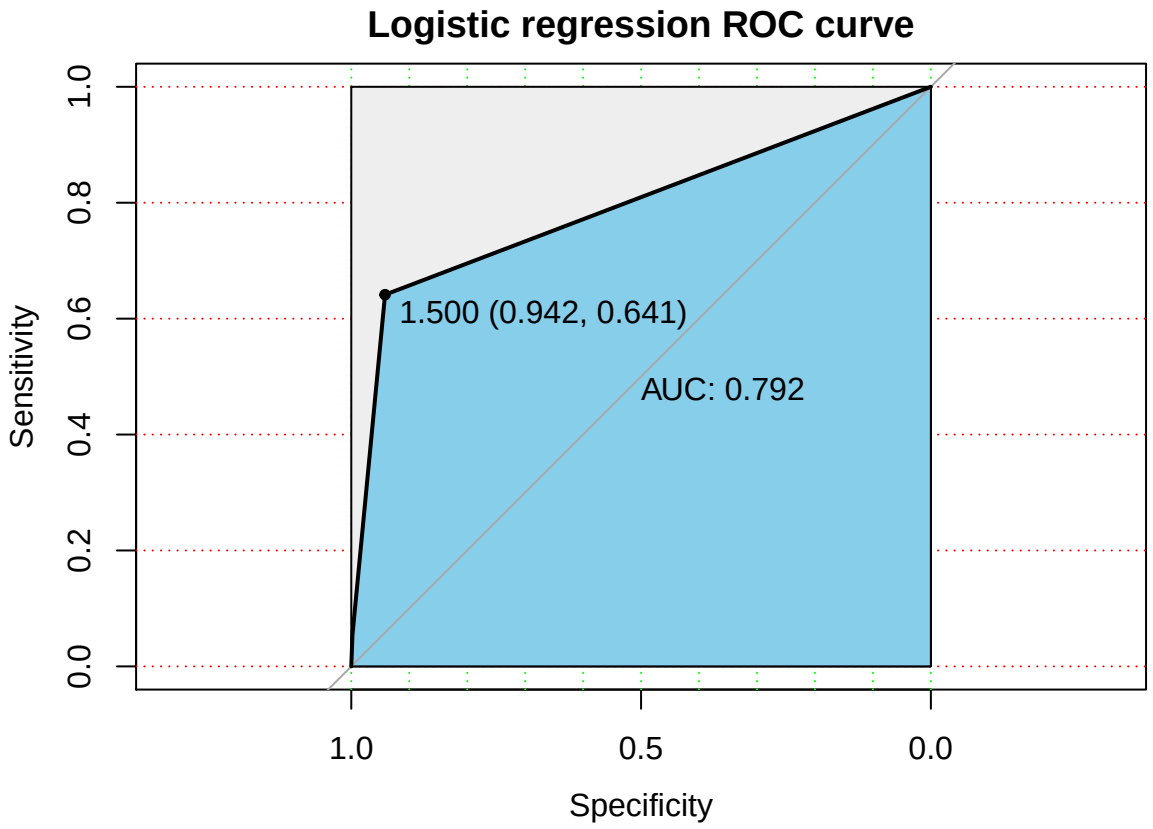
```
logi_roc <- roc(testing$fetal_health,as.numeric(pred_logi))
```

```
## Warning in roc.default(testing$fetal_health, as.numeric(pred_logi)): 'response'
## has more than two levels. Consider setting 'levels' explicitly or using
## 'multiclass.roc' instead
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
plot(logi_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "red"), ma
```



The predictions look to be pretty good in the sense that the confusion matrix is diagonal heavy with accuracy close to 90%. However, due to class imbalance problem with this dataset, we may want to look at sensitivity which is low for Class 2 and good for Class 1 and Class 3.The Auc is 0.792

Linear Discriminant Analysis (LDA)

Here, it was found that column 6 is constant through a group and it was decided to drop that variable since it will not be possible to fit LDA model with it.

```
library(MASS)
model_lda <- lda(fetal_health ~., training[,-6])
```

```
## Warning in lda.default(x, grouping, ...): variables are collinear
```

```
pred_lda <- predict(model_lda, testing)

confusionMatrix(testing$fetal_health, pred_lda$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 477  18   3
##          2  37  54   1
##          3   6   9  33
##
```

```
## Overall Statistics
##
##                Accuracy : 0.884
##                  95% CI : (0.8566, 0.9078)
##     No Information Rate : 0.815
##     P-Value [Acc > NIR] : 1.432e-06
##
##                   Kappa : 0.66
##
##  Mcnemar's Test P-Value : 0.002955
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9173  0.66667  0.89189
## Specificity            0.8220  0.93178  0.97504
## Pos Pred Value         0.9578  0.58696  0.68750
## Neg Pred Value         0.6929  0.95055  0.99322
## Prevalence             0.8150  0.12696  0.05799
## Detection Rate         0.7476  0.08464  0.05172
## Detection Prevalence   0.7806  0.14420  0.07524
## Balanced Accuracy      0.8697  0.79922  0.93347
```

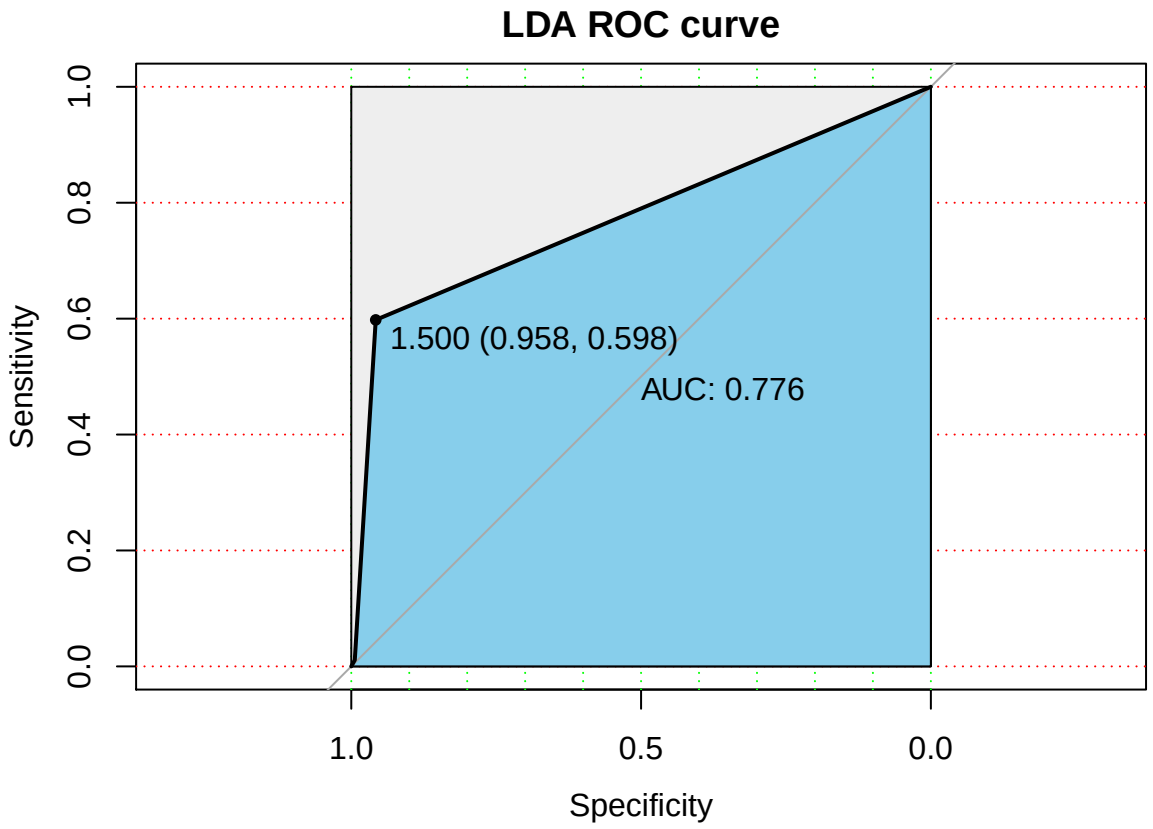
```
lda_roc <- roc(testing$fetal_health,as.numeric(pred_lda$class))
```

```
## Warning in roc.default(testing$fetal_health, as.numeric(pred_lda$class)):
## 'response' has more than two levels. Consider setting 'levels' explicitly or
## using 'multiclass.roc' instead
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
plot(lda_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "red"), max
```



The predictions from the LDA model is also decent, with accuracy close to 90% but a little bit lower than the multinomial regression. The sensitivity again is lower for class 2 but high for class 1 and class 3.The AUC is 0.776

Quadratic Discriminant Analysis(QDA)

Logistic regression and LDA are both for linear boundary classifier.To find out whether a linear or non-linear model is more appropriate here, we will try the quadratic discriminant model.

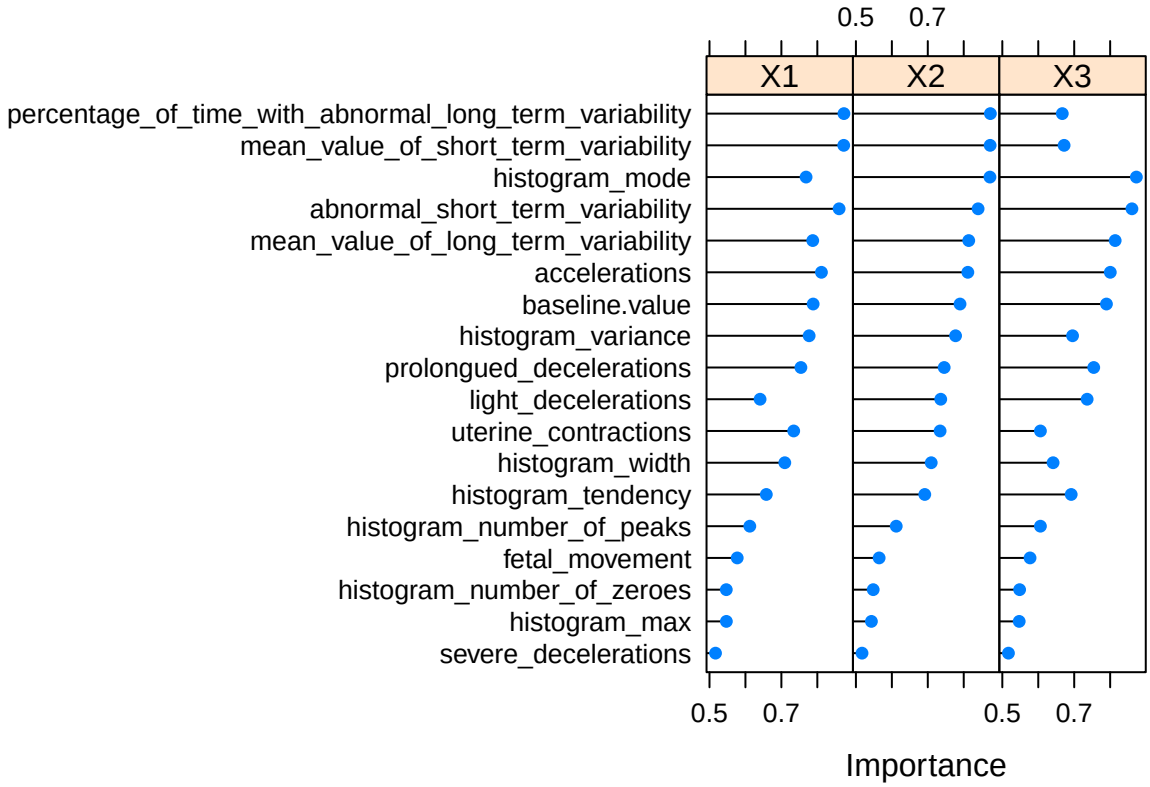
When simply run "model_qda <- qda(fetal_health ~.,training)",there will be an error of "rank deficiency in group 1". we suspect that this error might due to the shape of the dataset(too many indicator variables with insufficient samples),or due to the multicollinearity.so we need to select some indicator variables before we can run a QDA. We first delete the variables that are highly correlated and then use Learning Vector Quantization (LVQ) model to obtai the rank of attribute importance.

```
library(mlbench)
library(caret)
correlationMatrix <- cor(myfile[,1:21])
descrCorr <- cor(myfile[,1:21])
highCorr <- findCorrelation(descrCorr, 0.8)
myfile1 <- myfile[, -highCorr]
set.seed(123)
control <- trainControl(method="repeatedcv", number=10, repeats=3)
model <- train(fetal_health~., data=myfile1, method="lvq", preProcess="scale", trControl=control
```

```r
importance <- varImp(model, scale=FALSE)
importance
```

```
## ROC curve variable importance
##
##   variables are sorted by maximum importance across the classes
##                                                          X1     X2     X3
## percentage_of_time_with_abnormal_long_term_variability 0.8741 0.8741 0.6669
## mean_value_of_short_term_variability                   0.8734 0.8734 0.6719
## histogram_mode                                         0.7687 0.8729 0.8729
## abnormal_short_term_variability                        0.8606 0.8400 0.8606
## mean_value_of_long_term_variability                    0.7872 0.8139 0.8139
## accelerations                                          0.8114 0.8114 0.8004
## baseline.value                                         0.7884 0.7897 0.7897
## histogram_variance                                     0.7773 0.7773 0.6954
## prolongued_decelerations                               0.7543 0.7456 0.7543
## light_decelerations                                    0.6407 0.7360 0.7360
## uterine_contractions                                   0.7343 0.7343 0.6060
## histogram_width                                        0.7095 0.7095 0.6412
## histogram_tendency                                     0.6581 0.6916 0.6916
## histogram_number_of_peaks                              0.6122 0.6122 0.6060
## fetal_movement                                         0.5771 0.5646 0.5771
## histogram_number_of_zeroes                             0.5467 0.5481 0.5481
## histogram_max                                          0.5470 0.5431 0.5470
## severe_decelerations                                   0.5167 0.5170 0.5170
```

```r
plot(importance)
```




```
##run QDA model with 11 selected indicator variables
model_qda <- qda(fetal_health~percentage_of_time_with_abnormal_long_term_variability+mean_value_
##use the model to make prediction on testing data and see how the model performs
pred_qda <- predict(model_qda, testing)
confusionMatrix(testing$fetal_health, pred_qda$class)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 423  62  13
##          2  18  73   1
##          3   4  14  30
##
## Overall Statistics
##
##                Accuracy : 0.8245
##                  95% CI : (0.7927, 0.8532)
##     No Information Rate : 0.6975
##     P-Value [Acc > NIR] : 1.496e-13
##
##                   Kappa : 0.5787
##
##  Mcnemar's Test P-Value : 9.517e-09
##
## Statistics by Class:
```

```
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9506   0.4899  0.68182
## Specificity            0.6114   0.9611  0.96970
## Pos Pred Value         0.8494   0.7935  0.62500
## Neg Pred Value         0.8429   0.8608  0.97627
## Prevalence             0.6975   0.2335  0.06897
## Detection Rate         0.6630   0.1144  0.04702
## Detection Prevalence   0.7806   0.1442  0.07524
## Balanced Accuracy      0.7810   0.7255  0.82576
```

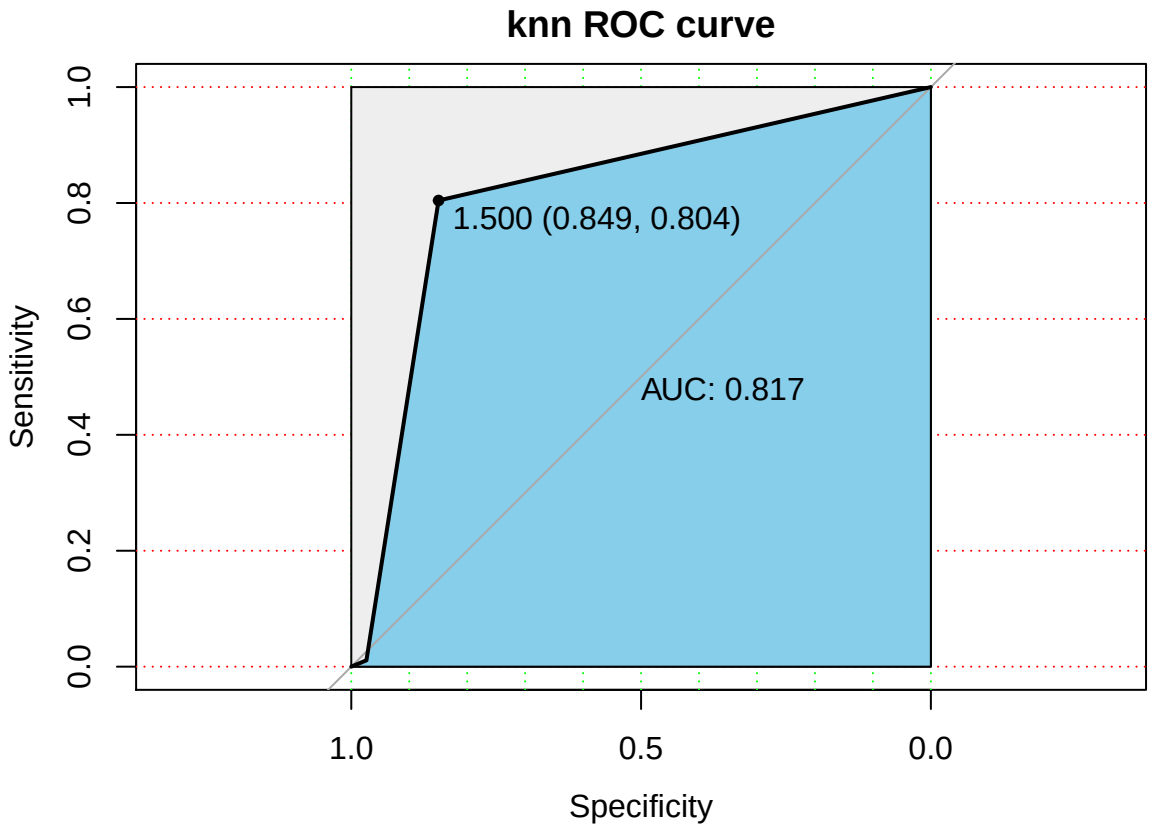
```
##ROC curve and  AUC value
QDA_roc <- roc(testing$fetal_health,as.numeric(pred_qda$class))
```

```
## Warning in roc.default(testing$fetal_health, as.numeric(pred_qda$class)):
## 'response' has more than two levels. Consider setting 'levels' explicitly or
## using 'multiclass.roc' instead
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
plot(QDA_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "red"), max
```



The accuracy is 0.8245, which is lower than those of linear model(logistic regression and LDA),that might because there is fewer indicator variables in QDA. But the AUC is 0.817 and is higher than those of linear model. Although we prefer the model with higher AUC, which means the

classification boundary is better because it closer to the Baye's one, due to the low out-of-sample accuracy and the problem of not able to having much indicator variables, we think QDA is not so appropriate for our data.

K-NearestNeighbor(KNN)

KNN is a non-parametric model and can be applied to the data without considering its classification boundary shape. But it requires the variables to be normalized.

```
##normalize the predictor variables
training.norm<-training
training.norm[, 1:21] <- sapply(training.norm[, 1:21],scale)
testing.norm<-testing
testing.norm[, 1:21] <- sapply(testing.norm[, 1:21],scale)
##try different k
model4_k3 <- knn(train = training.norm[, 1:21], test = testing.norm[, 1:21],
          cl = training.norm[, 22], k = 3)
confusionMatrix(model4_k3, testing.norm[, 22])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 477  31   8
##          2  19  57   4
##          3   2   4  36
##
## Overall Statistics
##
##                Accuracy : 0.8934
##                  95% CI : (0.8668, 0.9163)
##     No Information Rate : 0.7806
##     P-Value [Acc > NIR] : 6.956e-14
##
##                   Kappa : 0.6917
##
##  Mcnemar's Test P-Value : 0.09045
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9578  0.61957  0.75000
## Specificity            0.7214  0.95788  0.98983
## Pos Pred Value         0.9244  0.71250  0.85714
## Neg Pred Value         0.8279  0.93728  0.97987
## Prevalence             0.7806  0.14420  0.07524
## Detection Rate         0.7476  0.08934  0.05643
## Detection Prevalence   0.8088  0.12539  0.06583
## Balanced Accuracy      0.8396  0.78872  0.86992
```

```r
model4_k5 <- knn(train = training.norm[, 1:21], test = testing.norm[, 1:21],
          cl = training.norm[, 22], k = 5)
confusionMatrix(model4_k5, testing.norm[, 22])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 479  33   9
##          2  18  57   4
##          3   1   2  35
##
## Overall Statistics
##
##                Accuracy : 0.895
##                  95% CI : (0.8686, 0.9177)
##     No Information Rate : 0.7806
##     P-Value [Acc > NIR] : 2.913e-14
##
##                   Kappa : 0.6914
##
##  Mcnemar's Test P-Value : 0.009401
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9618  0.61957  0.72917
## Specificity            0.7000  0.95971  0.99492
## Pos Pred Value         0.9194  0.72152  0.92105
## Neg Pred Value         0.8376  0.93739  0.97833
## Prevalence             0.7806  0.14420  0.07524
## Detection Rate         0.7508  0.08934  0.05486
## Detection Prevalence   0.8166  0.12382  0.05956
## Balanced Accuracy      0.8309  0.78964  0.86204
```

```r
model4_k7 <- knn(train = training.norm[, 1:21], test = testing.norm[, 1:21],
          cl = training.norm[, 22], k = 7)
confusionMatrix(model4_k7, testing.norm[, 22])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 477  36   8
##          2  20  55   5
##          3   1   1  35
##
## Overall Statistics
```

```
##
##                Accuracy : 0.8887
##                  95% CI : (0.8617, 0.9121)
##     No Information Rate : 0.7806
##     P-Value [Acc > NIR] : 8.606e-13
##
##                   Kappa : 0.6728
##
##  Mcnemar's Test P-Value : 0.005376
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9578  0.59783  0.72917
## Specificity            0.6857  0.95421  0.99661
## Pos Pred Value         0.9155  0.68750  0.94595
## Neg Pred Value         0.8205  0.93369  0.97837
## Prevalence             0.7806  0.14420  0.07524
## Detection Rate         0.7476  0.08621  0.05486
## Detection Prevalence   0.8166  0.12539  0.05799
## Balanced Accuracy      0.8218  0.77602  0.86289
```

```
model4_k9 <- knn(train = training.norm[, 1:21], test = testing.norm[, 1:21],
          cl = training.norm[, 22], k = 9)
confusionMatrix(model4_k9, testing.norm[, 22])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 473  38   9
##          2  24  53   4
##          3   1   1  35
##
## Overall Statistics
##
##                Accuracy : 0.8793
##                  95% CI : (0.8515, 0.9036)
##     No Information Rate : 0.7806
##     P-Value [Acc > NIR] : 8.698e-11
##
##                   Kappa : 0.6462
##
##  Mcnemar's Test P-Value : 0.009924
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
```

```
## Sensitivity            0.9498  0.57609  0.72917
## Specificity            0.6643  0.94872  0.99661
## Pos Pred Value         0.9096  0.65432  0.94595
## Neg Pred Value         0.7881  0.92998  0.97837
## Prevalence             0.7806  0.14420  0.07524
## Detection Rate         0.7414  0.08307  0.05486
## Detection Prevalence   0.8150  0.12696  0.05799
## Balanced Accuracy      0.8070  0.76240  0.86289
```

```
##choose k=5 as the preferred model
model_knn <- knn(train = training.norm[, 1:21], test = testing.norm[, 1:21],
          cl = training.norm[, 22], k = 5)
confusionMatrix(model_knn, testing.norm[, 22])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 479  33   9
##          2  18  57   4
##          3   1   2  35
##
## Overall Statistics
##
##                Accuracy : 0.895
##                  95% CI : (0.8686, 0.9177)
##     No Information Rate : 0.7806
##     P-Value [Acc > NIR] : 2.913e-14
##
##                   Kappa : 0.6914
##
##  Mcnemar's Test P-Value : 0.009401
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.9618  0.61957  0.72917
## Specificity            0.7000  0.95971  0.99492
## Pos Pred Value         0.9194  0.72152  0.92105
## Neg Pred Value         0.8376  0.93739  0.97833
## Prevalence             0.7806  0.14420  0.07524
## Detection Rate         0.7508  0.08934  0.05486
## Detection Prevalence   0.8166  0.12382  0.05956
## Balanced Accuracy      0.8309  0.78964  0.86204
```

```
##ROC curve and  AUC value
knn_roc <- roc(testing.norm$fetal_health,as.numeric(model_knn))
```

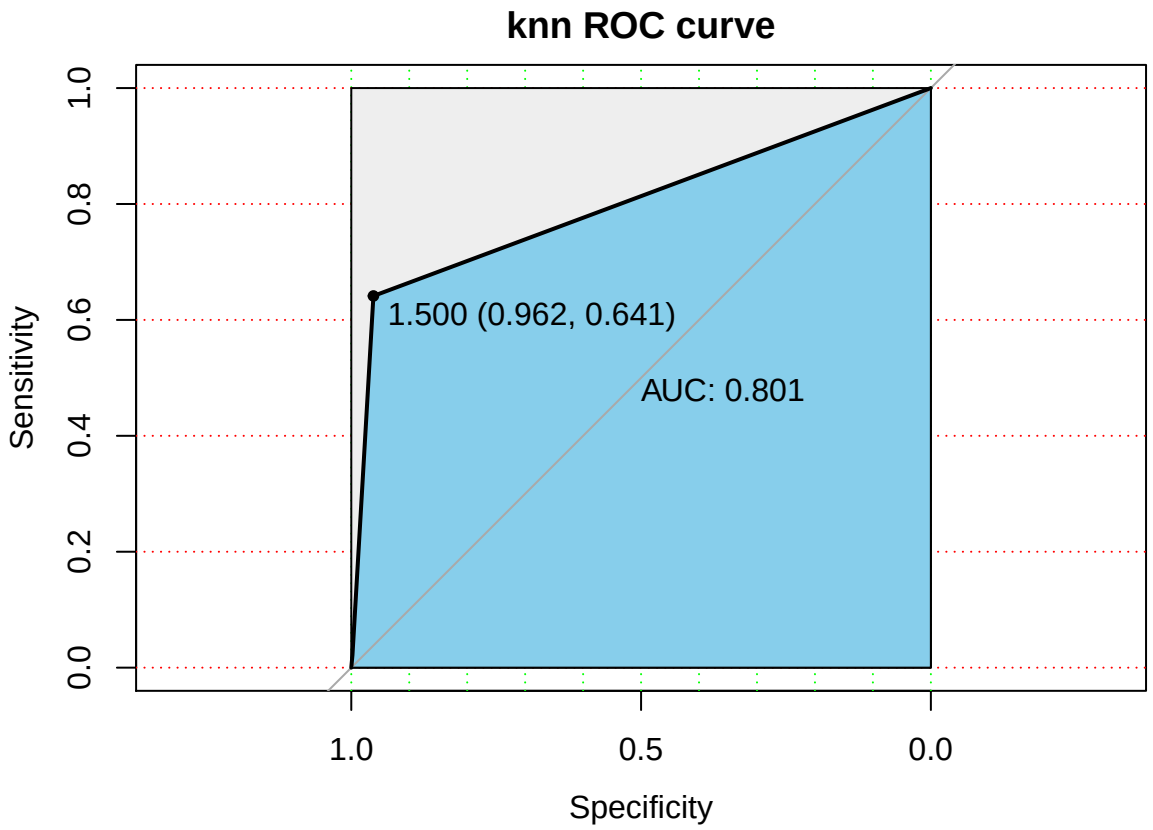
```
## Warning in roc.default(testing.norm$fetal_health, as.numeric(model_knn)):
```

```
## 'response' has more than two levels. Consider setting 'levels' explicitly or
## using 'multiclass.roc' instead
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

```
plot(knn_roc, print.auc=TRUE, auc.polygon=TRUE, grid=c(0.1, 0.2),grid.col=c("green", "red"), max
```



For KNN model,we tried different k values and chose k=5 as the preferred one. The model has a very nice accuracy of 89.5%,which is higher than the logistic regression's and the LDA's. The AUC is 0.801.

K-means clustering algorithm

K-means clustering algorithm is to classify the data into several clusters,so that these clusters have minimal within group variation(inter-cluster similarity). We would decide the number of clusters(k) by looking at the within group sum of squares of different clusters.

```
library(cluster)
library(rattle)
```

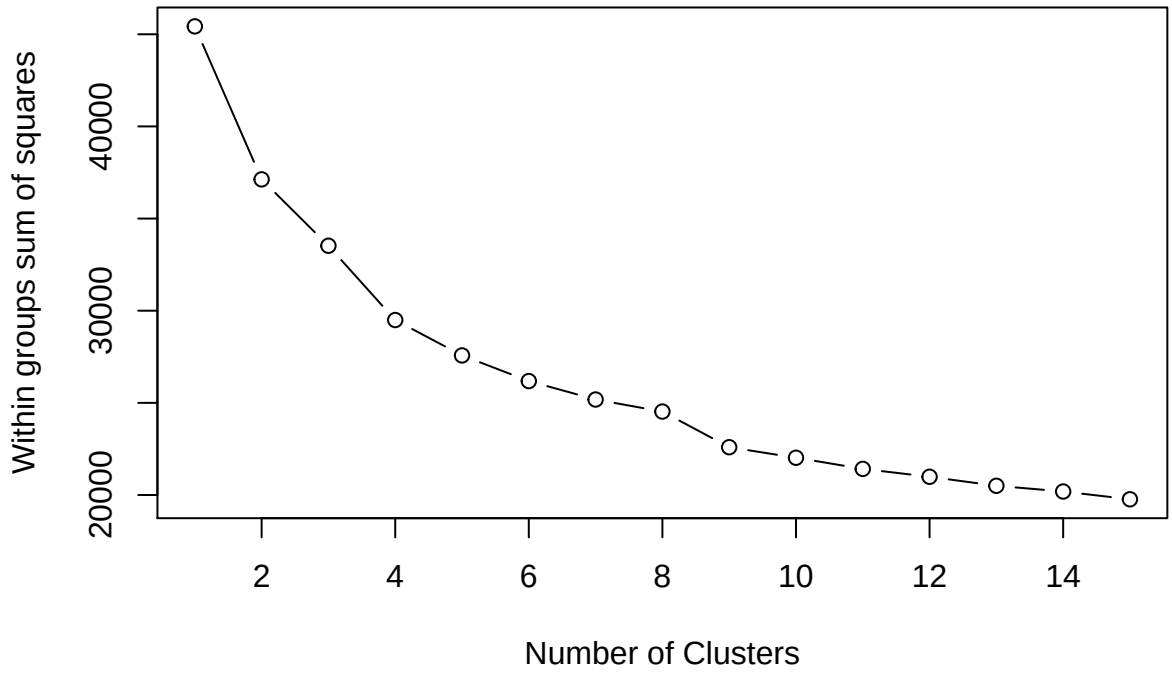
```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
```

```
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(NbClust)
myfile.norm<-myfile
myfile.norm[,1:21]<-sapply(myfile.norm[, 1:21],scale)
df_kmeans<-myfile.norm
library(NbClust)
set.seed(2)
wssplot <- function(data, nc=15, seed=1234){
                  wss <- (nrow(data)-1)*sum(apply(data,2,var))
                      for (i in 2:nc){
                set.seed(seed)
                    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
              plot(1:nc, wss, type="b", xlab="Number of Clusters",
                           ylab="Within groups sum of squares")
              wss
       }
wssplot(df_kmeans)
```




```
##  [1] 45427.10 37126.24 33524.59 29494.65 27573.33 26182.43 25180.99 24527.73
##  [9] 22593.26 22022.46 21417.73 20990.21 20500.42 20190.48 19768.23
```

```
## it is indicated that 4 clusters will be more appropriate
set.seed(2)
model_kmeans<- kmeans(df_kmeans, 4)
summary(model_kmeans)
```

```
##              Length Class  Mode
## cluster      2126   -none- numeric
## centers        88   -none- numeric
```

```
## totss           1   -none- numeric
## withinss        4   -none- numeric
## tot.withinss    1   -none- numeric
## betweenss       1   -none- numeric
## size            4   -none- numeric
## iter            1   -none- numeric
## ifault          1   -none- numeric
```

```
##show which cluster does each sample belong to
model_kmeans$cluster
```

```
##    [1] 2 4 4 4 4 1 1 2 2 2 4 4 2 4 1 1 4 1 1 1 1 1 2 3 3 3 3 1 1 4 1 4 4 4 4 4
##   [38] 4 4 4 2 2 2 4 4 4 4 4 4 4 4 3 4 1 3 3 4 4 3 3 4 3 3 3 4 4 3 4 3 3 3 3 4 4 4
##   [75] 4 3 3 4 4 4 4 4 4 4 4 4 4 4 3 4 3 3 3 3 3 3 3 3 3 4 2 2 2 2 2 4 4 4 4 2 4
##  [112] 4 4 2 4 1 1 2 2 2 4 2 2 4 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 4 4 4 4 4 4
##  [149] 3 3 3 3 4 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 2 2 2 2 2 2 4 2 4 2 4 4 4 2 4 4
##  [186] 4 2 3 3 3 4 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 4 4 2 2
##  [223] 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 2 2 2 2 4 4 3 3 3 3 4 2 2 2 4 4 4 4
##  [260] 4 2 2 4 2 2 3 4 2 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 2 2 2 3 3 3 3 3 3 3 4 4 3 3
##  [297] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3
##  [334] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 4 2 2 3 3 3 3 3 3 3 3 3
##  [371] 3 3 3 3 3 3 4 4 3 3 4 4 4 3 3 4 4 4 4 4 4 3 3 4 4 3 4 3 3 4 3 4 3 3 4 4 3
##  [408] 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 2 4 2 4 4 4 4 4 4 4 4 3 3
##  [445] 3 3 3 3 3 3 3 4 4 3 3 3 4 4 4 4 2 2 2 4 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4
##  [482] 3 3 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 4 4 4 4
##  [519] 4 4 2 2 3 3 3 3 4 4 3 4 4 4 4 3 3 4 3 3 3 3 4 3 4 3 3 3 3 3 3 3 3 4 4 4 4
##  [556] 4 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 2 2 2 2 2 2 2 2 4 4
##  [593] 4 2 3 3 3 3 3 3 3 3 3 3 4 3 3 3 4 4 4 4 4 4 4 4 4 4 3 4 4 3 4 4 4 4 4 4 2
##  [630] 4 4 4 3 3 4 3 3 4 4 4 4 4 4 4 4 3 3 2 3 2 3 3 3 3 3 3 4 4 2 2 2 2 1 4 1 4 4
##  [667] 3 3 4 4 4 3 3 4 4 4 3 4 4 4 3 4 1 1 2 4 2 4 4 1 1 4 4 1 4 1 4 1 1 1 1 1 1
##  [704] 1 1 4 3 2 2 2 2 4 2 4 3 3 3 3 2 3 3 3 3 3 3 3 3 2 2 2 2 2 4 4 4 4 2 2 2
##  [741] 2 4 3 4 1 4 4 4 3 3 3 3 3 3 3 3 2 2 4 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3
##  [778] 3 3 3 3 3 3 3 3 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 3 3 3 3 3 3 4 4 4
##  [815] 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 3
##  [852] 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3
##  [889] 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 3 4 4 4 3 4 4 3 2 2 2 2 2 2 2 2 2 2 2 2
##  [926] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 3 3 3 3 1 4 4 4 4 3 3 4 3 4 4 2 4 4 4 4 4 4 4
##  [963] 4 2 4 4 4 4 4 4 4 2 4 4 4 4 2 4 4 4 4 4 4 4 4 2 2 2 2 2 4 4 4 4 4 4 4 2 2 2 2 2 2
## [1000] 4 4 4 4 4 4 4 4 4 4 2 4 3 3 3 3 3 3 4 3 3 2 2 4 2 4 4 4 4 4 4 4 2 2 4 4 2 4 2
## [1037] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 4 4 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4
## [1074] 4 4 4 2 4 4 4 4 4 4 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1111] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 4 2 4 2 2 2 2 2 2 2 2 2 2
## [1148] 2 2 2 2 4 2 2 2 2 2 2 4 4 4 4 4 4 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1185] 2 2 2 2 2 2 2 2 2 2 4 3 3 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1222] 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [1259] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 4
## [1296] 2 4 4 4 4 4 4 3 3 3 3 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 4 4 4 1 4 4 4 4 4
## [1333] 4 4 1 4 1 4 1 4 4 4 4 4 4 4 1 1 1 1 1 2 2 4 2 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4
```

```
## [1370] 4 3 3 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4 3 3 3 3
## [1407] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 3 3 3 3 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1444] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 3
## [1481] 3 2 2 2 2 4 4 4 1 1 1 1 2 2 4 2 4 4 2 2 4 4 2 3 2 4 3 3 2 3 3 2 2 2 2 2 2
## [1518] 2 3 2 2 2 2 2 2 2 2 3 3 3 2 2 2 3 2 3 2 3 3 3 3 3 3 3 4 4 3 3 3 3 3 3 3 3
## [1555] 3 3 3 4 4 3 3 3 3 3 3 3 3 3 4 4 4 4 4 1 4 4 2 4 4 2 4 4 4 4 4 2 4 4 4 4 4 4
## [1592] 4 4 4 4 4 4 4 4 3 4 2 4 4 3 3 4 4 3 4 4 2 2 4 2 4 4 4 4 1 4 4 4 4 4 4 4 4
## [1629] 4 4 4 4 4 2 2 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 2 2 2 2 2 2 2 2 2 2 2
## [1666] 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1703] 4 3 3 3 3 3 1 4 4 4 2 2 4 4 4 4 4 4 2 2 2 4 4 4 4 1 4 4 4 4 4 4 4 4 4 4 4
## [1740] 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 3 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1777] 1 2 2 2 2 2 2 2 2 2 4 4 2 1 1 1 1 1 1 1 3 4 4 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [1814] 3 3 4 4 4 4 4 4 4 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1851] 4 4 4 4 4 4 4 3 3 3 3 3 3 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 3 3 3 3 3
## [1888] 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 4 4 3 4 4 4 3 4 4 4 4 4 4
## [1925] 4 4 4 1 4 2 4 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1962] 1 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 2
## [1999] 2 2 2 2 2 4 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
## [2036] 1 1 1 1 1 2 2 2 4 4 4 4 2 1 2 2 2 2 2 3 3 3 3 2 2 3 3 3 3 3 3 2 3 3 3 3 3
## [2073] 3 3 3 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 2 2 2 3 3 3 3 2
## [2110] 2 4 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3
```

```
##show the size of each cluster
model_kmeans$size
```

```
## [1] 154 575 723 674
```

```
##show the clusters produced by the model against the actual 3 types of fetal health
table(myfile.norm$fetal_health, model_kmeans$cluster)
```

```
##
##       1   2   3   4
##   1  35 558 419 643
##   2  10  12 242  31
##   3 109   5  62   0
```

```
## we can try to run a model with 3 clusters
## to see if the model can classify the data into 3 types based on the fetal health types.
model_kmeans1<- kmeans(testing.norm[,1:21], 3)
model_kmeans1$cluster<-factor(model_kmeans1$cluster)
confusionMatrix(model_kmeans1$cluster, testing.norm[, 22])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   1   2   3
##          1 139   5  25
##          2 171  75  13
##          3 188  12  10
##
```

```
## Overall Statistics
##
##                Accuracy : 0.3511
##                  95% CI : (0.314, 0.3895)
##     No Information Rate : 0.7806
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.086
##
##  Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3
## Sensitivity            0.2791   0.8152  0.20833
## Specificity            0.7857   0.6630  0.66102
## Pos Pred Value         0.8225   0.2896  0.04762
## Neg Pred Value         0.2345   0.9551  0.91121
## Prevalence             0.7806   0.1442  0.07524
## Detection Rate         0.2179   0.1176  0.01567
## Detection Prevalence   0.2649   0.4060  0.32915
## Balanced Accuracy      0.5324   0.7391  0.43468
```

```
kmeans_roc <- roc(testing.norm$fetal_health,as.numeric(model_kmeans1$cluster))
```

```
## Warning in roc.default(testing.norm$fetal_health,
## as.numeric(model_kmeans1$cluster)): 'response' has more than two levels.
## Consider setting 'levels' explicitly or using 'multiclass.roc' instead
```

```
## Setting levels: control = 1, case = 2
```

```
## Setting direction: controls < cases
```

Based on the result,this k-means model is not classifying the data so well based on the 3 types of fetal health, thus it is not suitable for our data and the objective of fetal health classification.

Conclusion

We fit our data with 5 models:logistic regression,LDA,QDA,KNN and K-means clustering. The first two model are linear, and the rest are non-linear(the shape of classification boundary does not matter with KNN and K-means). We use cross validation to evaluate the model performance,from which we could see the out-of-sample accuracy. Also we draw the ROC curve,from which we could see the AUC(area under curve).Comparing the results,we think the KNN model is the best fit for our data,with the accuracy of 0.895 and the AUC of 0.801,meaning that it has a nice predicting power on new data and it produces a classification boundary that is more closer to the Baye's one(the true boundary)

Reference

[1]Fetal Health Classification.https://www.kaggle.com/andrewmvd/fetal-health-classification?select=fetal_health

[2]https://rpubs.com/violetgirl/201598

[3]Randall R. Rojas. Econ412 lecture slides