Early Stage Diabetes model

Md Mushfiqur Rahman

9/6/2021

# R library

# Model  
library(caret)

load("ALL Model with Cluster.RData")

# Data Splitting (70% & 30%)

#data\_k2 <- data.frame(cbind(data\_clust, k\_clust$cluster))  
#names(data\_k2)[15] <- "Cluster"  
#data\_k10 <- data.frame(cbind(data\_clust, k\_clust\_10$cluster))  
#names(data\_k10)[15] <- "Cluster"  
  
#data\_k2$Cluster <- as.factor(data\_k2$Cluster)  
#data\_k10$Cluster <- as.factor(data\_k10$Cluster)  
######################### Creating training and testing data########################  
#ind\_train = createDataPartition(data\_k2$Cluster, p = 0.7, list = FALSE)  
#train <- data\_k2[ind\_train,]  
#test <- data\_k2[-ind\_train,]  
#train\_1 <- data\_k10[ind\_train, ]  
#test\_1 <- data\_k10[-ind\_train, ]

# Random forest ( for k=3)

#model\_RF <- train(Cluster ~ . , data = train, importance = TRUE, method='rf')  
  
predicted\_RF\_train <- predict(model\_RF, train)  
knitr::kable(table(predicted\_RF\_train, train$Cluster),caption = "Confusion Matrix for training dataset")

Confusion Matrix for training dataset

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 128 | 0 | 0 |
| 0 | 91 | 0 |
| 0 | 0 | 147 |

train\_cm\_RF <- confusionMatrix(train$Cluster, predicted\_RF\_train)  
  
predicted\_RF\_test <- predict(model\_RF, test)  
knitr:: kable(table(predicted\_RF\_test, test$Cluster), caption = "Confusion matrix for testing file")

Confusion matrix for testing file

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 54 | 0 | 0 |
| 0 | 38 | 0 |
| 0 | 0 | 62 |

test\_cm\_RF<-confusionMatrix(test$Cluster, predicted\_RF\_test)  
  
knitr::kable(data.frame(train\_cm\_RF$overall),caption = "Overall Accuracy Training dataset")

Overall Accuracy Training dataset

|  |  |
| --- | --- |
|  | train\_cm\_RF.overall |
| Accuracy | 1.0000000 |
| Kappa | 1.0000000 |
| AccuracyLower | 0.9899717 |
| AccuracyUpper | 1.0000000 |
| AccuracyNull | 0.4016393 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(test\_cm\_RF$overall), caption = "Overall Accuracy Testing dataset")

Overall Accuracy Testing dataset

|  |  |
| --- | --- |
|  | test\_cm\_RF.overall |
| Accuracy | 1.0000000 |
| Kappa | 1.0000000 |
| AccuracyLower | 0.9763309 |
| AccuracyUpper | 1.0000000 |
| AccuracyNull | 0.4025974 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(train\_cm\_RF$byClass),caption = "Other Measures Training dataset")

Other Measures Training dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.3497268 | 0.3497268 | 0.3497268 | 1 |
| Class: 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.2486339 | 0.2486339 | 0.2486339 | 1 |
| Class: 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.4016393 | 0.4016393 | 0.4016393 | 1 |

knitr::kable(data.frame(test\_cm\_RF$byClass), caption = "Other measures Testing dataset")

Other measures Testing dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.3506494 | 0.3506494 | 0.3506494 | 1 |
| Class: 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.2467532 | 0.2467532 | 0.2467532 | 1 |
| Class: 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.4025974 | 0.4025974 | 0.4025974 | 1 |

# Decision tree (k=3)

#model\_DT <- train(Cluster ~ . , data = train, method='rpart')  
  
predicted\_RF\_train <- predict(model\_RF, train)  
knitr::kable(table(predicted\_RF\_train, train$Cluster),caption = "Confusion Matrix for training dataset")

Confusion Matrix for training dataset

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 128 | 0 | 0 |
| 0 | 91 | 0 |
| 0 | 0 | 147 |

train\_cm\_RF <- confusionMatrix(train$Cluster, predicted\_RF\_train)  
  
predicted\_RF\_test <- predict(model\_RF, test)  
knitr:: kable(table(predicted\_RF\_test, test$Cluster), caption = "Confusion matrix for testing file")

Confusion matrix for testing file

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 54 | 0 | 0 |
| 0 | 38 | 0 |
| 0 | 0 | 62 |

test\_cm\_RF<-confusionMatrix(test$Cluster, predicted\_RF\_test)  
  
knitr::kable(data.frame(train\_cm\_RF$overall),caption = "Overall Accuracy Training dataset")

Overall Accuracy Training dataset

|  |  |
| --- | --- |
|  | train\_cm\_RF.overall |
| Accuracy | 1.0000000 |
| Kappa | 1.0000000 |
| AccuracyLower | 0.9899717 |
| AccuracyUpper | 1.0000000 |
| AccuracyNull | 0.4016393 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(test\_cm\_RF$overall), caption = "Overall Accuracy Testing dataset")

Overall Accuracy Testing dataset

|  |  |
| --- | --- |
|  | test\_cm\_RF.overall |
| Accuracy | 1.0000000 |
| Kappa | 1.0000000 |
| AccuracyLower | 0.9763309 |
| AccuracyUpper | 1.0000000 |
| AccuracyNull | 0.4025974 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(train\_cm\_RF$byClass),caption = "Other Measures Training dataset")

Other Measures Training dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.3497268 | 0.3497268 | 0.3497268 | 1 |
| Class: 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.2486339 | 0.2486339 | 0.2486339 | 1 |
| Class: 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.4016393 | 0.4016393 | 0.4016393 | 1 |

knitr::kable(data.frame(test\_cm\_RF$byClass), caption = "Other measures Testing dataset")

Other measures Testing dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.3506494 | 0.3506494 | 0.3506494 | 1 |
| Class: 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.2467532 | 0.2467532 | 0.2467532 | 1 |
| Class: 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.4025974 | 0.4025974 | 0.4025974 | 1 |

# Support Vector Machine (Radial Kurnel) (K=3)

library(kernlab) # SVM methodology

##   
## Attaching package: 'kernlab'

## The following object is masked from 'package:ggplot2':  
##   
## alpha

library(e1071) # SVM methodology  
  
#model\_SVM <- train(Cluster~., data=train, method='lssvmRadial')  
  
predicted\_SVM\_train <- predict(model\_SVM, train)  
knitr::kable(table(predicted\_SVM\_train, train$Cluster),caption = "Confusion Matrix for training dataset")

Confusion Matrix for training dataset

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 121 | 1 | 0 |
| 4 | 90 | 0 |
| 3 | 0 | 147 |

train\_cm\_SVM<-confusionMatrix(predicted\_SVM\_train, train$Cluster)  
  
predicted\_SVM\_test <- predict(model\_SVM, test)  
knitr:: kable(table(predicted\_SVM\_test, test$Cluster), caption = "Confusion matrix for testing file")

Confusion matrix for testing file

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 50 | 0 | 0 |
| 2 | 38 | 0 |
| 2 | 0 | 62 |

test\_cm\_SVM<-confusionMatrix(predicted\_SVM\_test, test$Cluster)  
  
knitr::kable(data.frame(train\_cm\_SVM$overall),caption = "Overall Accuracy training dataset")

Overall Accuracy training dataset

|  |  |
| --- | --- |
|  | train\_cm\_SVM.overall |
| Accuracy | 0.9781421 |
| Kappa | 0.9666272 |
| AccuracyLower | 0.9573870 |
| AccuracyUpper | 0.9905172 |
| AccuracyNull | 0.4016393 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(test\_cm\_SVM$overall), caption = "Overall Accuracy testing dataset")

Overall Accuracy testing dataset

|  |  |
| --- | --- |
|  | test\_cm\_SVM.overall |
| Accuracy | 0.9740260 |
| Kappa | 0.9603297 |
| AccuracyLower | 0.9348336 |
| AccuracyUpper | 0.9928786 |
| AccuracyNull | 0.4025974 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(train\_cm\_SVM$byClass),caption = "Other Measures training dataset")

Other Measures training dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 0.9453125 | 0.9957983 | 0.9918033 | 0.9713115 | 0.9918033 | 0.9453125 | 0.968000 | 0.3497268 | 0.3306011 | 0.3333333 | 0.9705554 |
| Class: 2 | 0.9890110 | 0.9854545 | 0.9574468 | 0.9963235 | 0.9574468 | 0.9890110 | 0.972973 | 0.2486339 | 0.2459016 | 0.2568306 | 0.9872328 |
| Class: 3 | 1.0000000 | 0.9863014 | 0.9800000 | 1.0000000 | 0.9800000 | 1.0000000 | 0.989899 | 0.4016393 | 0.4016393 | 0.4098361 | 0.9931507 |

knitr::kable(data.frame(test\_cm\_SVM$byClass), caption = "Other measures testing dataset")

Other measures testing dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 0.9259259 | 1.0000000 | 1.00000 | 0.9615385 | 1.00000 | 0.9259259 | 0.9615385 | 0.3506494 | 0.3246753 | 0.3246753 | 0.9629630 |
| Class: 2 | 1.0000000 | 0.9827586 | 0.95000 | 1.0000000 | 0.95000 | 1.0000000 | 0.9743590 | 0.2467532 | 0.2467532 | 0.2597403 | 0.9913793 |
| Class: 3 | 1.0000000 | 0.9782609 | 0.96875 | 1.0000000 | 0.96875 | 1.0000000 | 0.9841270 | 0.4025974 | 0.4025974 | 0.4155844 | 0.9891304 |

# Multi-Layer Perception

#model\_MLP <- train(Cluster~., data=train, method='mlp')  
  
predicted\_MLP\_train <- predict(model\_MLP, train)  
knitr::kable(table(predicted\_MLP\_train, train$Cluster),caption = "Confusion Matrix for training dataset")

Confusion Matrix for training dataset

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 128 | 0 | 0 |
| 0 | 91 | 0 |
| 0 | 0 | 147 |

train\_cm\_MLP<-confusionMatrix(predicted\_MLP\_train, train$Cluster)  
  
predicted\_MLP\_test <- predict(model\_MLP, test)  
knitr:: kable(table(predicted\_MLP\_test, test$Cluster), caption = "Confusion matrix for testing file")

Confusion matrix for testing file

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 53 | 0 | 0 |
| 1 | 37 | 0 |
| 0 | 1 | 62 |

test\_cm\_MLP<-confusionMatrix(predicted\_MLP\_test, test$Cluster)  
  
knitr::kable(data.frame(train\_cm\_MLP$overall),caption = "Overall Accuracy training dataset")

Overall Accuracy training dataset

|  |  |
| --- | --- |
|  | train\_cm\_MLP.overall |
| Accuracy | 1.0000000 |
| Kappa | 1.0000000 |
| AccuracyLower | 0.9899717 |
| AccuracyUpper | 1.0000000 |
| AccuracyNull | 0.4016393 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(test\_cm\_MLP$overall), caption = "Overall Accuracy testing dataset")

Overall Accuracy testing dataset

|  |  |
| --- | --- |
|  | test\_cm\_MLP.overall |
| Accuracy | 0.9870130 |
| Kappa | 0.9801342 |
| AccuracyLower | 0.9538755 |
| AccuracyUpper | 0.9984233 |
| AccuracyNull | 0.4025974 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | NaN |

knitr::kable(data.frame(train\_cm\_MLP$byClass),caption = "Other Measures training dataset")

Other Measures training dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.3497268 | 0.3497268 | 0.3497268 | 1 |
| Class: 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.2486339 | 0.2486339 | 0.2486339 | 1 |
| Class: 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.4016393 | 0.4016393 | 0.4016393 | 1 |

knitr::kable(data.frame(test\_cm\_MLP$byClass), caption = "Other measures testing dataset")

Other measures testing dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 0.9814815 | 1.0000000 | 1.0000000 | 0.9900990 | 1.0000000 | 0.9814815 | 0.9906542 | 0.3506494 | 0.3441558 | 0.3441558 | 0.9907407 |
| Class: 2 | 0.9736842 | 0.9913793 | 0.9736842 | 0.9913793 | 0.9736842 | 0.9736842 | 0.9736842 | 0.2467532 | 0.2402597 | 0.2467532 | 0.9825318 |
| Class: 3 | 1.0000000 | 0.9891304 | 0.9841270 | 1.0000000 | 0.9841270 | 1.0000000 | 0.9920000 | 0.4025974 | 0.4025974 | 0.4090909 | 0.9945652 |

# K-Nearest Neighbours (k=3)

#model\_KNN <- train(Cluster ~ ., data = train, method = "knn", preProcess = c("center","scale"), tuneLength = 20 )  
  
  
predicted\_KNN\_train <- predict(model\_KNN, train)  
knitr::kable(table(predicted\_KNN\_train, train$Cluster),caption = "Confusion Matrix for training dataset")

Confusion Matrix for training dataset

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 125 | 3 | 1 |
| 0 | 87 | 5 |
| 3 | 1 | 141 |

train\_cm\_KNN<-confusionMatrix(predicted\_KNN\_train, train$Cluster)  
  
predicted\_KNN\_test <- predict(model\_KNN, test)  
knitr:: kable(table(predicted\_KNN\_test, test$Cluster), caption = "Confusion matrix for testing file")

Confusion matrix for testing file

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 51 | 0 | 0 |
| 1 | 37 | 2 |
| 2 | 1 | 60 |

test\_cm\_KNN<-confusionMatrix(predicted\_KNN\_test, test$Cluster)  
  
knitr::kable(data.frame(train\_cm\_KNN$overall),caption = "Overall Accuracy training dataset")

Overall Accuracy training dataset

|  |  |
| --- | --- |
|  | train\_cm\_KNN.overall |
| Accuracy | 0.9644809 |
| Kappa | 0.9457821 |
| AccuracyLower | 0.9400270 |
| AccuracyUpper | 0.9809545 |
| AccuracyNull | 0.4016393 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | 0.0833163 |

knitr::kable(data.frame(test\_cm\_KNN$overall), caption = "Overall Accuracy testing dataset")

Overall Accuracy testing dataset

|  |  |
| --- | --- |
|  | test\_cm\_KNN.overall |
| Accuracy | 0.9610390 |
| Kappa | 0.9405252 |
| AccuracyLower | 0.9171295 |
| AccuracyUpper | 0.9855700 |
| AccuracyNull | 0.4025974 |
| AccuracyPValue | 0.0000000 |
| McnemarPValue | 0.3430301 |

knitr::kable(data.frame(train\_cm\_KNN$byClass),caption = "Other Measures training dataset")

Other Measures training dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 0.9765625 | 0.9831933 | 0.9689922 | 0.9873418 | 0.9689922 | 0.9765625 | 0.9727626 | 0.3497268 | 0.3415301 | 0.3524590 | 0.9798779 |
| Class: 2 | 0.9560440 | 0.9818182 | 0.9456522 | 0.9854015 | 0.9456522 | 0.9560440 | 0.9508197 | 0.2486339 | 0.2377049 | 0.2513661 | 0.9689311 |
| Class: 3 | 0.9591837 | 0.9817352 | 0.9724138 | 0.9728507 | 0.9724138 | 0.9591837 | 0.9657534 | 0.4016393 | 0.3852459 | 0.3961749 | 0.9704594 |

knitr::kable(data.frame(test\_cm\_KNN$byClass), caption = "Other measures testing dataset")

Other measures testing dataset

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Sensitivity | Specificity | Pos.Pred.Value | Neg.Pred.Value | Precision | Recall | F1 | Prevalence | Detection.Rate | Detection.Prevalence | Balanced.Accuracy |
| Class: 1 | 0.9444444 | 1.0000000 | 1.000000 | 0.9708738 | 1.000000 | 0.9444444 | 0.9714286 | 0.3506494 | 0.3311688 | 0.3311688 | 0.9722222 |
| Class: 2 | 0.9736842 | 0.9741379 | 0.925000 | 0.9912281 | 0.925000 | 0.9736842 | 0.9487179 | 0.2467532 | 0.2402597 | 0.2597403 | 0.9739111 |
| Class: 3 | 0.9677419 | 0.9673913 | 0.952381 | 0.9780220 | 0.952381 | 0.9677419 | 0.9600000 | 0.4025974 | 0.3896104 | 0.4090909 | 0.9675666 |

### ROC for Test data

library("ROCR")  
library(pROC)

# Testing data  
##  
roc\_RF\_test <- as.numeric(predict(model\_RF, test, type = 'raw'))  
roc\_RF\_test1 <- multiclass.roc(test$Cluster,roc\_RF\_test )

## Setting direction: controls < cases

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

rs\_RF\_test <- roc\_RF\_test1[['rocs']]  
  
##  
roc\_MLP\_test <- as.numeric(predict(model\_MLP, test, type = 'raw'))  
roc\_MLP\_test1 <- multiclass.roc(test$Cluster,roc\_MLP\_test )

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

rs\_MLP\_test <- roc\_MLP\_test1[['rocs']]  
  
##  
roc\_DT\_test <- as.numeric(predict(model\_DT, test, type = 'raw'))  
roc\_DT\_test1 <- multiclass.roc(test$Cluster,roc\_DT\_test )

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

rs\_DT\_test <- roc\_DT\_test1[['rocs']]  
  
##  
roc\_SVM\_test <- as.numeric(predict(model\_SVM, test, type = 'raw'))  
roc\_SVM\_test1 <- multiclass.roc(test$Cluster,roc\_SVM\_test )

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

rs\_SVM\_test <- roc\_SVM\_test1[['rocs']]  
##  
roc\_KNN\_test <- as.numeric(predict(model\_KNN, test, type = 'raw'))  
roc\_KNN\_test1 <- multiclass.roc(test$Cluster,roc\_KNN\_test )

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

rs\_KNN\_test <- roc\_KNN\_test1[['rocs']]  
  
  
# AUC Test  
auc(roc\_RF\_test1)

## Multi-class area under the curve: 1

auc(roc\_DT\_test1)

## Multi-class area under the curve: 0.9443

auc(roc\_SVM\_test1)

## Multi-class area under the curve: 0.9753

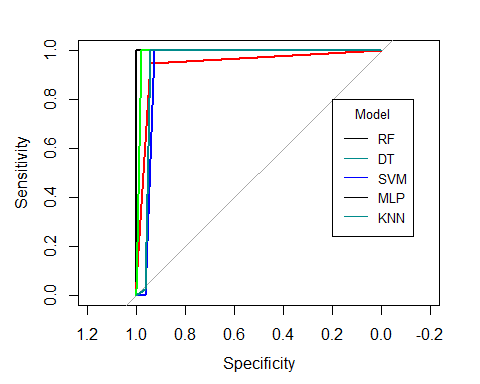
auc(roc\_MLP\_test1)

## Multi-class area under the curve: 0.9926

auc(roc\_KNN\_test1)

## Multi-class area under the curve: 0.9686

par(mfrow=c(1,1))  
# ROC test  
plot.roc(rs\_RF\_test[[1]])  
plot.roc(rs\_DT\_test[[1]], add=TRUE,col= "red") #1  
plot.roc(rs\_SVM\_test[[1]], add=TRUE, col= "blue")   
plot.roc(rs\_MLP\_test[[1]], add= TRUE, col="green") #1  
plot.roc(rs\_KNN\_test[[1]], add= TRUE, col="cyan4")#1  
legend(0.2,0.8 , legend=c("RF", "DT","SVM","MLP","KNN"),  
 col=c("Black", "cyan4","blue"), lty=c(1,1,1), cex=0.8,  
 title="Model", text.font=1)



## ROC Train data

roc\_RF\_train <- as.numeric(predict(model\_RF, train, type = 'raw'))  
roc\_RF\_train1 <- multiclass.roc(train$Cluster,roc\_RF\_train )

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

rs\_RF\_train <- roc\_RF\_train1[['rocs']]  
  
##  
roc\_MLP\_train <- as.numeric(predict(model\_MLP, train, type = 'raw'))  
roc\_MLP\_train1 <- multiclass.roc(train$Cluster,roc\_MLP\_train )

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

rs\_MLP\_train <- roc\_MLP\_train1[['rocs']]  
  
##  
roc\_DT\_train <- as.numeric(predict(model\_DT, train, type = 'raw'))  
roc\_DT\_train1 <- multiclass.roc(train$Cluster,roc\_DT\_train )

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

rs\_DT\_train <- roc\_DT\_train1[['rocs']]  
  
##  
roc\_SVM\_train <- as.numeric(predict(model\_SVM, train, type = 'raw'))  
roc\_SVM\_train1 <- multiclass.roc(train$Cluster,roc\_SVM\_train )

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

rs\_SVM\_train <- roc\_SVM\_train1[['rocs']]  
##  
roc\_KNN\_train <- as.numeric(predict(model\_KNN, train, type = 'raw'))  
roc\_KNN\_train1 <- multiclass.roc(train$Cluster,roc\_KNN\_train )

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

rs\_KNN\_train <- roc\_KNN\_train1[['rocs']]  
  
  
# AUC train  
auc(roc\_RF\_train1)

## Multi-class area under the curve: 1

auc(roc\_DT\_train1)

## Multi-class area under the curve: 0.945

auc(roc\_SVM\_train1)

## Multi-class area under the curve: 0.9813

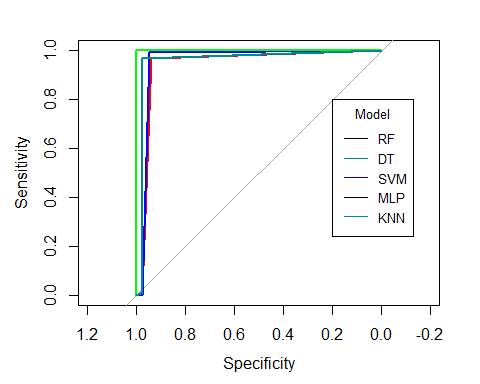
auc(roc\_MLP\_train1)

## Multi-class area under the curve: 1

auc(roc\_KNN\_train1)

## Multi-class area under the curve: 0.9688

par(mfrow=c(1,1))  
# ROC train  
plot.roc(rs\_RF\_train[[1]])  
plot.roc(rs\_DT\_train[[1]], add=TRUE,col= "red") #1  
plot.roc(rs\_SVM\_train[[1]], add=TRUE, col= "blue")   
plot.roc(rs\_MLP\_train[[1]], add= TRUE, col="green") #1  
plot.roc(rs\_KNN\_train[[1]], add= TRUE, col="cyan4")#1  
legend(0.2,0.8 , legend=c("RF", "DT","SVM","MLP","KNN"),  
 col=c("Black", "cyan4","blue"), lty=c(1,1,1), cex=0.8,  
 title="Model", text.font=1)



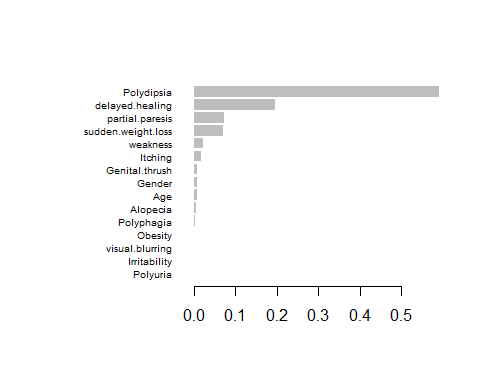
# XGboost feature Importance

# XGBOOST   
suppressPackageStartupMessages({  
 library(xgboost);  
 library(shapr );  
 library(Matrix);  
 library(SHAPforxgboost)})  
  
  
sparse\_matrix <- sparse.model.matrix(Cluster ~ .-1, data = data\_k2)  
output\_vector = data\_k2$Cluster  
xg\_df<- xgb.DMatrix(data = sparse\_matrix, label = output\_vector )  
  
df\_x <- as.matrix(data\_k2[,-17])  
df\_y <- as.matrix(data\_k2[ , 17])  
  
param\_list <- list(objective = "reg:squarederror",  
 eta = 0.02,  
 max\_depth = 10,  
 gamma = 0.01,  
 subsample = 0.95  
)  
xgb\_model <- xgboost::xgboost(data = df\_x,   
 label = df\_y,   
 params = param\_list, nrounds = 10,  
 verbose = FALSE, nthread = parallel::detectCores() - 2,  
 early\_stopping\_rounds = 8)  
  
# compute feature importance matrix  
  
importance\_matrix = xgb.importance(feature\_names = colnames(df\_x),  
 model = xgb\_model)  
knitr::kable(data.frame(importance\_matrix), caption = "Variable Importance with XGboost")

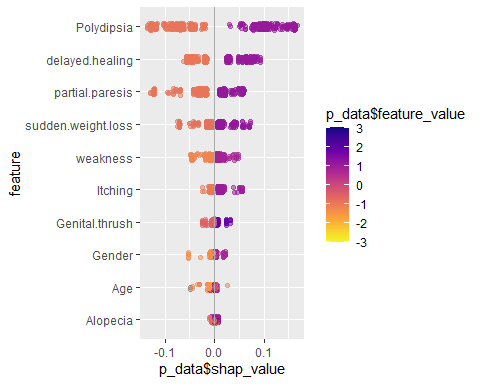
Variable Importance with XGboost

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Gain | Cover | Frequency |
| Polydipsia | 0.5915270 | 0.3293969 | 0.0662252 |
| delayed.healing | 0.1964927 | 0.1821457 | 0.0728477 |
| partial.paresis | 0.0721256 | 0.1495863 | 0.0728477 |
| sudden.weight.loss | 0.0691589 | 0.0503069 | 0.0728477 |
| weakness | 0.0220670 | 0.0445690 | 0.1523179 |
| Itching | 0.0166757 | 0.0926741 | 0.0927152 |
| Genital.thrush | 0.0078460 | 0.0326928 | 0.0529801 |
| Gender | 0.0075471 | 0.0109421 | 0.0728477 |
| Age | 0.0071372 | 0.0497732 | 0.1986755 |
| Alopecia | 0.0037883 | 0.0078730 | 0.0397351 |
| Polyphagia | 0.0035468 | 0.0099413 | 0.0529801 |
| Obesity | 0.0006123 | 0.0010008 | 0.0066225 |
| visual.blurring | 0.0005455 | 0.0006005 | 0.0066225 |
| Irritability | 0.0004871 | 0.0012010 | 0.0066225 |
| Polyuria | 0.0004428 | 0.0372965 | 0.0331126 |

# XGBoost Importance plot  
xgb.plot.importance(importance\_matrix = importance\_matrix)



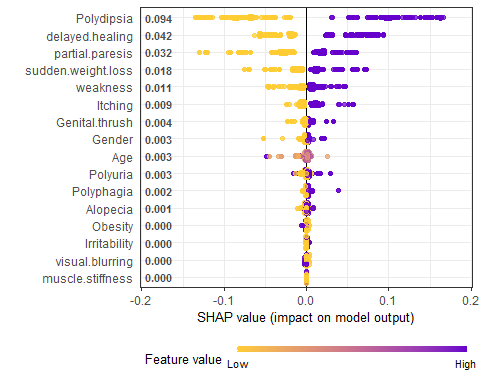
# Shap  
shap\_values <- shap.values(xgb\_model, xg\_df)  
xgb.ggplot.shap.summary(df\_x, model = xgb\_model)



shap\_values$mean\_shap\_score

## Polydipsia delayed.healing partial.paresis sudden.weight.loss   
## 0.0938993023 0.0422095291 0.0319326819 0.0183039085   
## weakness Itching Genital.thrush Gender   
## 0.0106696619 0.0088489557 0.0037295621 0.0034789454   
## Age Polyuria Polyphagia Alopecia   
## 0.0030508709 0.0030007084 0.0018069677 0.0010354167   
## Obesity Irritability visual.blurring muscle.stiffness   
## 0.0002888584 0.0001867122 0.0001861276 0.0000000000

# To prepare the long-format data:  
shap\_long <- shap.prep(shap\_contrib = shap\_values$shap\_score, X\_train = df\_x)  
# \*\*SHAP summary plot\*\*  
shap.plot.summary(shap\_long)



# Dependence Plot for top 6 features  
fig\_list <- lapply(names(shap\_values$mean\_shap\_score)[1:10],   
 shap.plot.dependence, data\_long = shap\_long)  
gridExtra::grid.arrange(grobs = fig\_list, ncol = 5)

