R Notebook

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
library(readr)
library(nnet)
library(ISLR)
library(e1071)
library(ROSE)
library(randomForest)
library(caret)
```

```
set.seed(123)

data = read_csv("archive (1)/healthcare-dataset-stroke-data.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   id = col_double(),
##   gender = col_character(),
##   age = col_double(),
##   hypertension = col_double(),
##   heart_disease = col_double(),
##   ever_married = col_character(),
##   work_type = col_character(),
##   Residence_type = col_character(),
##   avg_glucose_level = col_double(),
##   bmi = col_character(),
##   smoking_status = col_character(),
##   stroke = col_double()
## )
```

```
data$bmi[data$bmi == "N/A"] = NA
data$missing_bmi = as.factor(is.na(data$bmi))
data$gender[data$gender == "Other"] = "Female"
data$gender = as.factor(data$gender)
data$age = as.numeric(data$age)
data$hypertension = as.factor(data$hypertension)
data$heart_disease = as.factor(data$heart_disease)
data$ever_married = as.factor(data$ever_married)
data$work_type = as.factor(data$work_type)
data$Residence_type = as.factor(data$Residence_type)
```

```r
data$bmi = as.numeric(data$bmi)
data$smoking_status = as.factor(data$smoking_status)
data$stroke = as.factor(data$stroke)

summary(data)
```

```
##        id           gender          age         hypertension heart_disease
##  Min.   :   67   Female:2995   Min.   : 0.08   0:4612       0:4834
##  1st Qu.:17741   Male  :2115   1st Qu.:25.00   1: 498       1: 276
##  Median :36932                 Median :45.00
##  Mean   :36518                 Mean   :43.23
##  3rd Qu.:54682                 3rd Qu.:61.00
##  Max.   :72940                 Max.   :82.00
##
##  ever_married          work_type    Residence_type avg_glucose_level
##  No :1757     children     : 687   Rural:2514     Min.   : 55.12
##  Yes:3353     Govt_job     : 657   Urban:2596     1st Qu.: 77.25
##               Never_worked :  22                  Median : 91.89
##               Private      :2925                  Mean   :106.15
##               Self-employed: 819                  3rd Qu.:114.09
##                                                   Max.   :271.74
##
##       bmi                 smoking_status stroke   missing_bmi
##  Min.   :10.30   formerly smoked: 885   0:4861   FALSE:4909
##  1st Qu.:23.50   never smoked   :1892   1: 249   TRUE : 201
##  Median :28.10   smokes         : 789
##  Mean   :28.89   Unknown        :1544
##  3rd Qu.:33.10
##  Max.   :97.60
##  NA's   :201
```

```r
# Gives the BMI the predicted value
BMIFit = glm(bmi ~ gender + age + hypertension + heart_disease + ever_married + work_type + Residence_ty
BMIPredictions = predict(BMIFit, newdata = data)
s = is.na(data$bmi)
data$bmi[s] = BMIPredictions[s]
```

```r
smp_size <- floor(0.8 * nrow(data))
train_ind <- sample(seq_len(nrow(data)), size = smp_size)
train <- data[train_ind, ]
test <- data[-train_ind, ]
```

```r
# Fit logistic regression on training set

strokeFit = glm(stroke ~ . - id, data = train, family = binomial)
strokeTrainPred = predict(strokeFit, newdata = train, type = "response")
t = 0
maxAcc = 0
for (i in 0:100) {
  strokePredLabels = as.numeric(strokeTrainPred > i/100)
  acc = mean(train$stroke == strokePredLabels)
  if (acc > maxAcc) {
    t = i/100
    maxAcc = acc
  }
}

summary(strokeFit)
```

```
##
## Call:
## glm(formula = stroke ~ . - id, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6735  -0.2962  -0.1528  -0.0845   3.2792
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -6.310589   0.813107  -7.761 8.42e-15 ***
## genderMale                  -0.072987   0.164039  -0.445  0.65637
## age                          0.075476   0.006799  11.101  < 2e-16 ***
## hypertension1                0.420485   0.187240   2.246  0.02472 *
## heart_disease1               0.259486   0.219162   1.184  0.23642
## ever_marriedYes             -0.141760   0.258787  -0.548  0.58384
## work_typeGovt_job           -1.504253   0.882884  -1.704  0.08842 .
## work_typeNever_worked      -10.609923 341.541807  -0.031  0.97522
## work_typePrivate            -1.324579   0.861319  -1.538  0.12409
## work_typeSelf-employed      -1.715397   0.888649  -1.930  0.05356 .
## Residence_typeUrban         -0.017579   0.159393  -0.110  0.91218
## avg_glucose_level            0.004295   0.001374   3.125  0.00178 **
## bmi                         -0.003055   0.014101  -0.217  0.82849
## smoking_statusnever smoked  -0.149961   0.200640  -0.747  0.45481
## smoking_statussmokes         0.053940   0.249312   0.216  0.82871
## smoking_statusUnknown       -0.272542   0.247048  -1.103  0.26994
## missing_bmiTRUE              1.454805   0.240168   6.057 1.38e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1555.2  on 4087  degrees of freedom
## Residual deviance: 1197.0  on 4071  degrees of freedom
## AIC: 1231
##
## Number of Fisher Scoring iterations: 14
```

```
# Creates the oversampled data then makes a test/train split

data2 <- ovun.sample(stroke~.,data = data, method = ’over’,p = 0.3)$data
sample_index <- sample(nrow(data2),nrow(data2)*0.8)
train2 <- data2[sample_index,]
test2 <- data2[-sample_index,]
summary(data2)
```

```
##        id           gender          age        hypertension heart_disease
##  Min.   :   67   Female:4024   Min.   : 0.08   0:5949       0:6314
##  1st Qu.:17873   Male  :2902   1st Qu.:32.00   1: 977       1: 612
##  Median :36831                 Median :54.00
##  Mean   :36755                 Mean   :49.76
##  3rd Qu.:55302                 3rd Qu.:70.00
##  Max.   :72940                 Max.   :82.00
##  ever_married         work_type    Residence_type avg_glucose_level
##  No :1946     children     : 699   Rural:3322     Min.   : 55.12
##  Yes:4980     Govt_job     : 890   Urban:3604     1st Qu.: 77.61
##               Never_worked :  22                  Median : 94.00
##               Private      :4025                  Mean   :113.18
##               Self-employed:1290                  3rd Qu.:124.63
##                                                   Max.   :271.74
##       bmi                 smoking_status stroke   missing_bmi
##  Min.   :10.30   formerly smoked:1371   0:4861   FALSE:6418
##  1st Qu.:24.50   never smoked   :2549   1:2065   TRUE : 508
##  Median :28.70   smokes         :1086
##  Mean   :29.28   Unknown        :1920
##  3rd Qu.:32.80
##  Max.   :97.60
```

```r
# Makes the rf model on the training data
forest1 <- randomForest(stroke~.-id,data = train2,ntree = 500,mtry = 3)
forest1
```

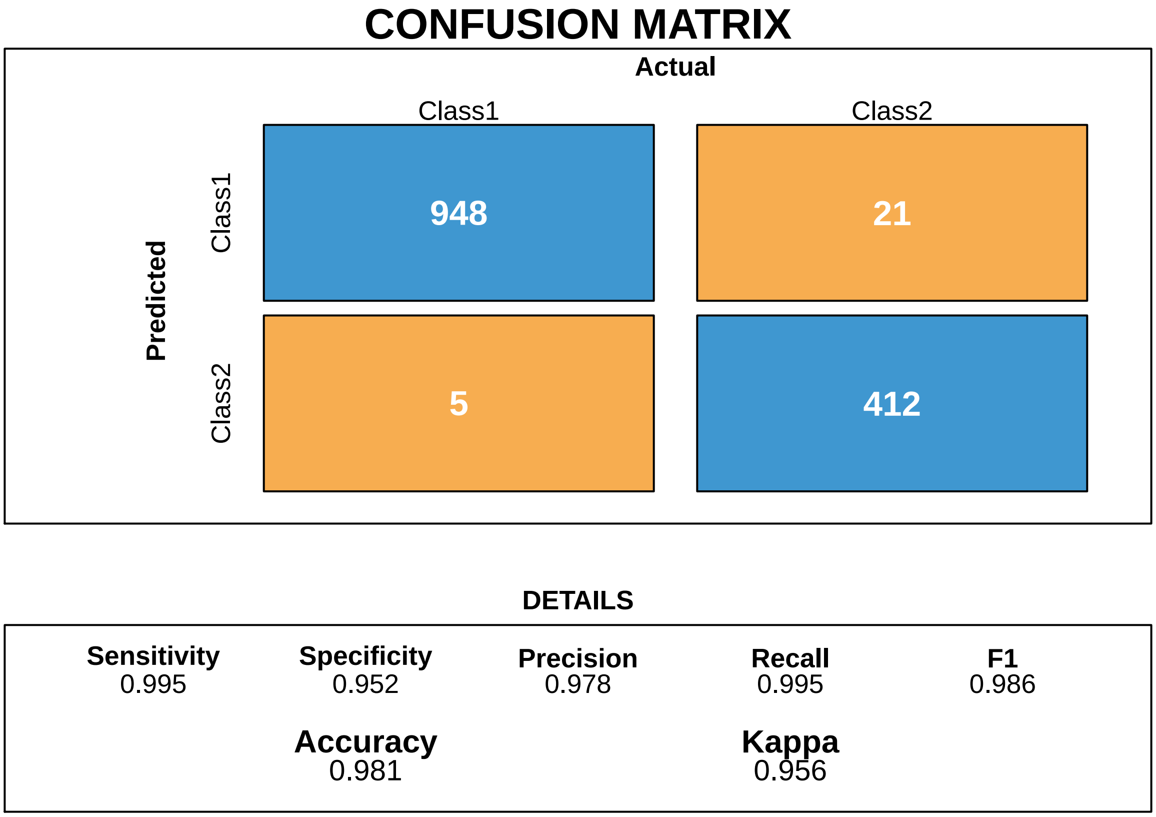
```
##
## Call:
##  randomForest(formula = stroke ~ . - id, data = train2, ntree = 500,      mtry = 3)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 1.39%
## Confusion matrix:
##      0    1 class.error
## 0 3818   74 0.019013361
## 1    3 1645 0.001820388
```

```r
draw_confusion_matrix <- function(cm) {

  layout(matrix(c(1,1,2)))
  par(mar=c(2,2,2,2))
  plot(c(100, 345), c(300, 450), type = "n", xlab="", ylab="", xaxt='n', yaxt='n')
  title('CONFUSION MATRIX', cex.main=2)

  # create the matrix
  rect(150, 430, 240, 370, col='#3F97D0')
  text(195, 435, 'Class1', cex=1.2)
  rect(250, 430, 340, 370, col='#F7AD50')
  text(295, 435, 'Class2', cex=1.2)
  text(125, 370, 'Predicted', cex=1.3, srt=90, font=2)
  text(245, 450, 'Actual', cex=1.3, font=2)
  rect(150, 305, 240, 365, col='#F7AD50')
  rect(250, 305, 340, 365, col='#3F97D0')
  text(140, 400, 'Class1', cex=1.2, srt=90)
  text(140, 335, 'Class2', cex=1.2, srt=90)

  # add in the cm results
  res <- as.numeric(cm$table)
  text(195, 400, res[1], cex=1.6, font=2, col='white')
  text(195, 335, res[2], cex=1.6, font=2, col='white')
  text(295, 400, res[3], cex=1.6, font=2, col='white')
  text(295, 335, res[4], cex=1.6, font=2, col='white')

  # add in the specifics
  plot(c(100, 0), c(100, 0), type = "n", xlab="", ylab="", main = "DETAILS", xaxt='n', yaxt='n')
  text(10, 85, names(cm$byClass[1]), cex=1.2, font=2)
  text(10, 70, round(as.numeric(cm$byClass[1]), 3), cex=1.2)
  text(30, 85, names(cm$byClass[2]), cex=1.2, font=2)
  text(30, 70, round(as.numeric(cm$byClass[2]), 3), cex=1.2)
  text(50, 85, names(cm$byClass[5]), cex=1.2, font=2)
  text(50, 70, round(as.numeric(cm$byClass[5]), 3), cex=1.2)
  text(70, 85, names(cm$byClass[6]), cex=1.2, font=2)
  text(70, 70, round(as.numeric(cm$byClass[6]), 3), cex=1.2)
  text(90, 85, names(cm$byClass[7]), cex=1.2, font=2)
  text(90, 70, round(as.numeric(cm$byClass[7]), 3), cex=1.2)

  # add in the accuracy information
  text(30, 35, names(cm$overall[1]), cex=1.5, font=2)
  text(30, 20, round(as.numeric(cm$overall[1]), 3), cex=1.4)
  text(70, 35, names(cm$overall[2]), cex=1.5, font=2)
  text(70, 20, round(as.numeric(cm$overall[2]), 3), cex=1.4)
}
```

```
g = predict(forest1, newdata = test2)
cf = confusionMatrix(test2$stroke, g)
cf
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 948  21
##          1   5 412
##
##                Accuracy : 0.9812
##                  95% CI : (0.9726, 0.9877)
##     No Information Rate : 0.6876
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9559
##
##  Mcnemar's Test P-Value : 0.003264
##
##             Sensitivity : 0.9948
##             Specificity : 0.9515
##          Pos Pred Value : 0.9783
##          Neg Pred Value : 0.9880
##              Prevalence : 0.6876
##          Detection Rate : 0.6840
##    Detection Prevalence : 0.6991
##       Balanced Accuracy : 0.9731
##
##        'Positive' Class : 0
##
```

```
draw_confusion_matrix(cf)
```



CONFUSION MATRIX

		Actual	
		Class1	Class2
Predicted	Class1	948	21
	Class2	5	412

DETAILS

Sensitivity	Specificity	Precision	Recall	F1
0.995	0.952	0.978	0.995	0.986

Accuracy	Kappa
0.981	0.956