Classification

Tyler Frankenberg

3/20/2022

Import packages

```
library(tidyverse)
library(pROC)
library(caret)
```

Import and examine data

```
url <- "https://raw.githubusercontent.com/curdferguson/data621/main/datasets/classification-outp
ut-data.csv"

class_raw <- read_csv(url, col_names = TRUE)

head(class_raw)
```

```
## # A tibble: 6 x 11
##   pregnant glucose diastolic skinfold insulin   bmi pedigree   age class
##      <dbl>   <dbl>     <dbl>    <dbl>   <dbl> <dbl>    <dbl> <dbl> <dbl>
## 1        7     124        70       33     215  25.5    0.161    37     0
## 2        2     122        76       27     200  35.9    0.483    26     0
## 3        3     107        62       13      48  22.9    0.678    23     1
## 4        1      91        64       24       0  29.2    0.192    21     0
## 5        4      83        86       19       0  29.3    0.317    34     0
## 6        1     100        74       12      46  19.5    0.149    28     0
## # ... with 2 more variables: scored.class <dbl>, scored.probability <dbl>
```

```
summary(class_raw)
```

```
##     pregnant         glucose        diastolic        skinfold
##  Min.   : 0.000   Min.   : 57.0   Min.   : 38.0   Min.   : 0.0
##  1st Qu.: 1.000   1st Qu.: 99.0   1st Qu.: 64.0   1st Qu.: 0.0
##  Median : 3.000   Median :112.0   Median : 70.0   Median :22.0
##  Mean   : 3.862   Mean   :118.3   Mean   : 71.7   Mean   :19.8
##  3rd Qu.: 6.000   3rd Qu.:136.0   3rd Qu.: 78.0   3rd Qu.:32.0
##  Max.   :15.000   Max.   :197.0   Max.   :104.0   Max.   :54.0
##     insulin            bmi           pedigree           age
##  Min.   :  0.00   Min.   :19.40   Min.   :0.0850   Min.   :21.00
##  1st Qu.:  0.00   1st Qu.:26.30   1st Qu.:0.2570   1st Qu.:24.00
##  Median :  0.00   Median :31.60   Median :0.3910   Median :30.00
##  Mean   : 63.77   Mean   :31.58   Mean   :0.4496   Mean   :33.31
##  3rd Qu.:105.00   3rd Qu.:36.00   3rd Qu.:0.5800   3rd Qu.:41.00
##  Max.   :543.00   Max.   :50.00   Max.   :2.2880   Max.   :67.00
##      class         scored.class    scored.probability
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.02323
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.11702
##  Median :0.0000   Median :0.0000   Median :0.23999
##  Mean   :0.3149   Mean   :0.1768   Mean   :0.30373
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.43093
##  Max.   :1.0000   Max.   :1.0000   Max.   :0.94633
```

Contingency Function

```
contingency <- function(df) {
    df1 <- data_frame(df)
    colnames(df1) <- c("class", "scored.class", "scored.prob")

    ct <- with(df1, table(df1$class, df1$scored.class, dnn= c("class", "scored.class")))

    cat("Contingency Table:", "\n")
    cat("\n")
    print(ct)

    tp <- ct[2, 2]
    tn <- ct[1, 1]
    fp <- ct[1, 2]
    fn <- ct[2, 1]
    total <- sum(ct)

    acc <- (tp + tn) / total
    err <- (fp + fn) / total


    cat("\n")
    cat("Accuracy:  ", acc, "\n")
    cat("Classification Error Rate: ", err, "\n")

    if (acc + err != 1) {
      cat("\n")
      cat("...")
      cat("ERROR: Accuracy and Error do not sum to 1")
      break
    } else {

      prec <- tp / (tp + fp)
      sens <- tp / (tp + fn)
      spec <- tn / (tn + fp)
      f1 <- (2 * prec * sens) / (prec + sens)

      cat("...", "\n")
      cat("Precision:   ", prec, "\n")
      cat("Sensitivity: ", sens, "\n")
      cat("Specificity: ", spec, "\n")
      cat("F1 Score:    ", f1, "\n")
    }
}

class_df <- data_frame(as_factor(class_raw$class),
                       as_factor(class_raw$scored.class),
                  class_raw$scored.probability)

colnames(class_df) <- c("class", "scored.class", "scored.prob")

contingency(class_df)
```

```
## Contingency Table:
##
##      scored.class
## class   0   1
##     0 119   5
##     1  30  27
##
## Accuracy:   0.8066298
## Classification Error Rate:  0.1933702
## ...
## Precision:    0.84375
## Sensitivity:  0.4736842
## Specificity:  0.9596774
## F1 Score:     0.6067416
```

```
confusionMatrix(class_df$scored.class, reference=class_df$class)
```

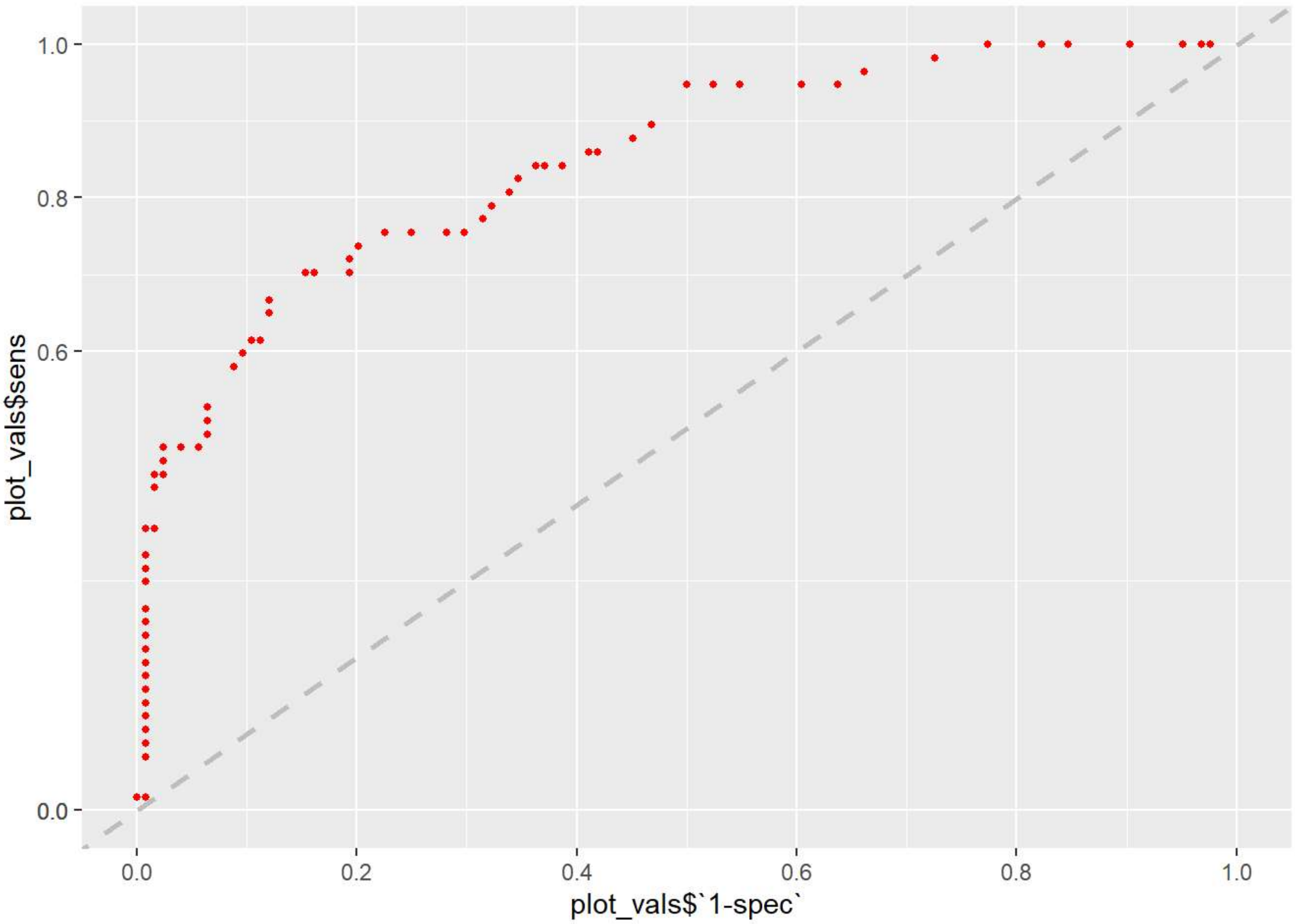
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 119  30
##          1   5  27
##
##                Accuracy : 0.8066
##                  95% CI : (0.7415, 0.8615)
##     No Information Rate : 0.6851
##     P-Value [Acc > NIR] : 0.0001712
##
##                   Kappa : 0.4916
##
##  Mcnemar's Test P-Value : 4.976e-05
##
##             Sensitivity : 0.9597
##             Specificity : 0.4737
##          Pos Pred Value : 0.7987
##          Neg Pred Value : 0.8438
##              Prevalence : 0.6851
##          Detection Rate : 0.6575
##    Detection Prevalence : 0.8232
##       Balanced Accuracy : 0.7167
##
##        'Positive' Class : 0
##
```

ROC Curve Function

```
df2 <- data_frame(class_df)
colnames(df2) <- c("class", "scored.class", "scored.prob")

scored.class_list <- lapply(seq(0.03, 0.94, 0.01), function(i) {
  i = if_else(df2$scored.prob >= i, 1, 0) })

ct_list <- lapply(scored.class_list, function(j) {
  table(df2$class, j, dnn= c("class", paste0("scored.class_")))
})

tp_list <- lapply(seq(1, 92, 1), function(k) {
  ct_list[[k]][4]
})

tn_list <- lapply(seq(1, 92, 1), function(k) {
  ct_list[[k]][1]
})

fp_list <- lapply(seq(1, 92, 1), function(k) {
  ct_list[[k]][3]
})

fn_list <- lapply(seq(1, 92, 1), function(k) {
  ct_list[[k]][2]
})

plot_vals <- data_frame(index = seq(3, 94, 1),
                        threshold = seq(0.03, 0.94, 0.01),
                        tp = unlist(tp_list),
                        tn = unlist(tn_list),
                        fp = unlist(fp_list),
                        fn = unlist(fn_list),
                        sens = tp / (tp + fn),
                        spec = tn / (tn + fp),
                        `1-spec` = 1 - spec)




roc_plot <- ggplot(plot_vals, aes(x=plot_vals$`1-spec`, y=plot_vals$sens)) +
        geom_point(col="red", size = 1) +
        scale_x_continuous(limits=c(0.0, 1.0), breaks=c(0, 0.20, 0.40, 0.60, 0.80, 1)) +
        scale_y_continuous(limits=c(0.0, 1.0), breaks=c(0, 0,20, 0,40, 0.60, 0.80, 1)) +
        geom_abline(slope = 1, intercept = 0, col="gray", size=1, linetype="dashed")

roc_plot
```

```
## Warning: Use of `plot_vals$`1-spec`` is discouraged. Use `1-spec` instead.
```

```
## Warning: Use of `plot_vals$sens` is discouraged. Use `sens` instead.
```



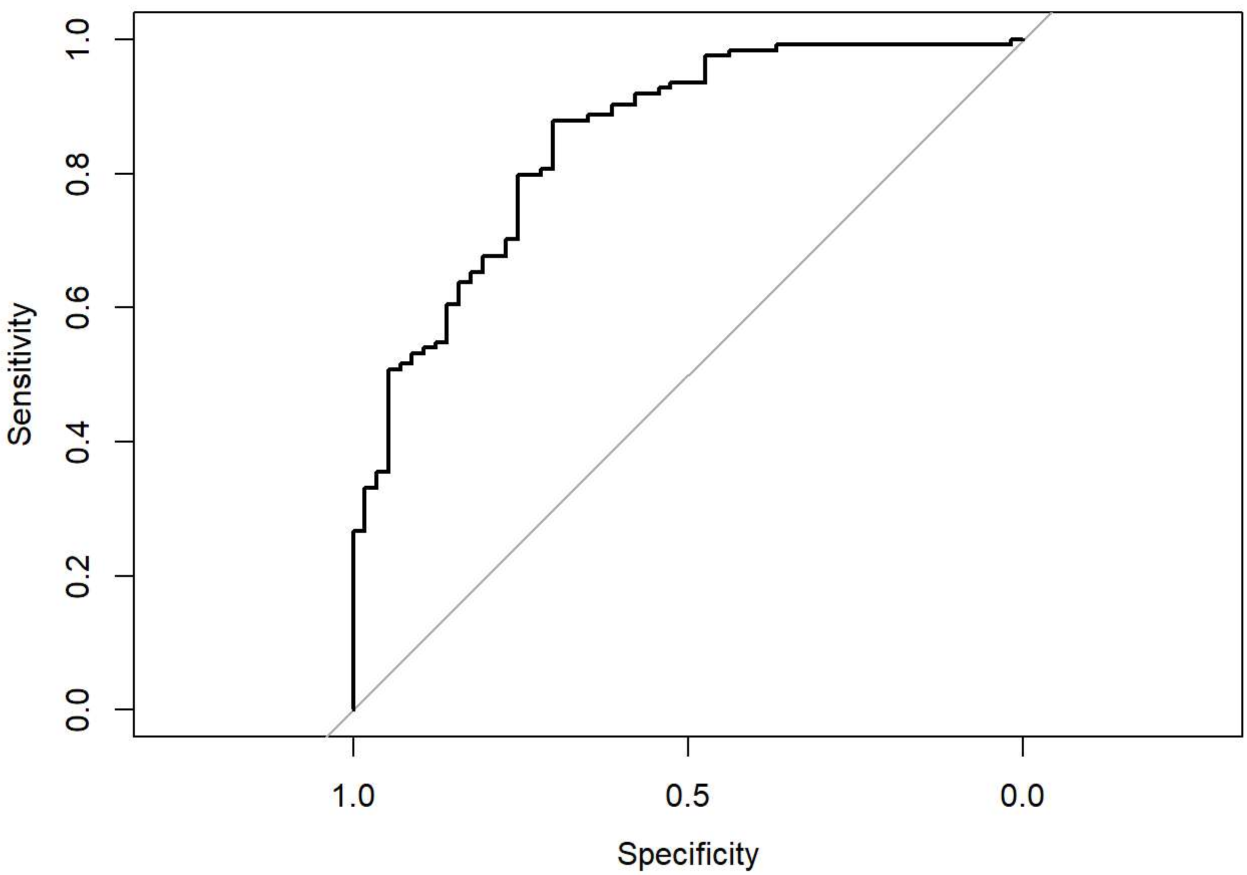
```
rocCurve <- roc(class_df, response= class,
                predictor= scored.prob, levels = rev(levels(class_df$class)))
```

```
## Setting direction: controls > cases
```

```
auc(rocCurve)
```

```
## Area under the curve: 0.8503
```

```
plot.roc(rocCurve)
```


Sensitivity
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.5
0.0
Specificity