ROC Curve

Wei Chieh Chen

2025-04-12

Data preparation, GLM model, Cutoffs:

The code loads the body dataset, subsets it to Gender, Height, and Weight, converts Gender to a factor, and fits a logistic regression model. The fitted probabilities are stored in fitted.GLM_model. A sequence of cutoff values is created, and the code uses sapply to create a prediction matrix where each column corresponds to the predictions using one cutoff value.

```
# ----------------------------
# 1. Data preparation and GLM model
# ----------------------------
# Load the data
data(body, package="gclus")
# names(body)
# str(body)
# head(body)

# Subset to only the three variables and convert Gender to a factor.
body <- within(body[, c("Weight", "Height",  "Gender")], {
  Gender <- as.factor(Gender)
  })

# Fit the GLM (logistic regression) using Height and Weight as predictors.
GLM_model <- glm(Gender ~ Weight +Height, family=binomial(logit), data=body)
summary(GLM_model)
```

```
##
## Call:
## glm(formula = Gender ~ Weight + Height, family = binomial(logit),
##     data = body)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.99092    4.19014  -9.783  < 2e-16 ***
## Weight        0.10149    0.01611   6.298 3.01e-10 ***
## Height        0.19846    0.02537   7.821 5.23e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 702.52  on 506  degrees of freedom
## Residual deviance: 340.41  on 504  degrees of freedom
## AIC: 346.41
```

```
##
## Number of Fisher Scoring iterations: 6
```

```
# Extract fitted probabilities.
body<- within(body, {
  fitted.GLM_model <- fitted(GLM_model)
})

# ----------------------------
# 2. Generate Predictions for a Sequence of Cutoffs
# ----------------------------
# Create a sequence of cutoff values.
thresholds <- seq(0, 1, 0.001)

# Using sapply to compute predictions for each threshold.
# Each column in pred_matrix corresponds to 1*(fitted.GLM_model > cutoff) for one cutoff.
pred_matrix <- sapply(thresholds, function(cutoff) 1 * (body$fitted.GLM_model > cutoff))

# ----------------------------
# 3. Compute Sensitivity and Specificity for each cutoff
# ----------------------------
# Convert actual Gender values to binary indicator.
# Assuming the second level of Gender is the "positive" class.
actual <- ifelse(body$Gender == levels(body$Gender)[2], 1, 0)

# Sensitivity (TP / (TP + FN)) and Specificity (TN / (TN + FP))
sensitivity <- sapply(1:ncol(pred_matrix), function(i) {
  preds <- pred_matrix[, i]
  TP <- sum(preds == 1 & actual == 1)
  FN <- sum(preds == 0 & actual == 1)
  if ((TP + FN) == 0) NA else TP / (TP + FN)
})

specificity <- sapply(1:ncol(pred_matrix), function(i) {
  preds <- pred_matrix[, i]
  TN <- sum(preds == 0 & actual == 0)
  FP <- sum(preds == 1 & actual == 0)
  if ((TN + FP) == 0) NA else TN / (TN + FP)
})

# Combine into a data frame.
roc_data <- data.frame(Threshold = thresholds, Specificity = specificity, Sensitivity = sensitivity)
```

ROC Curve

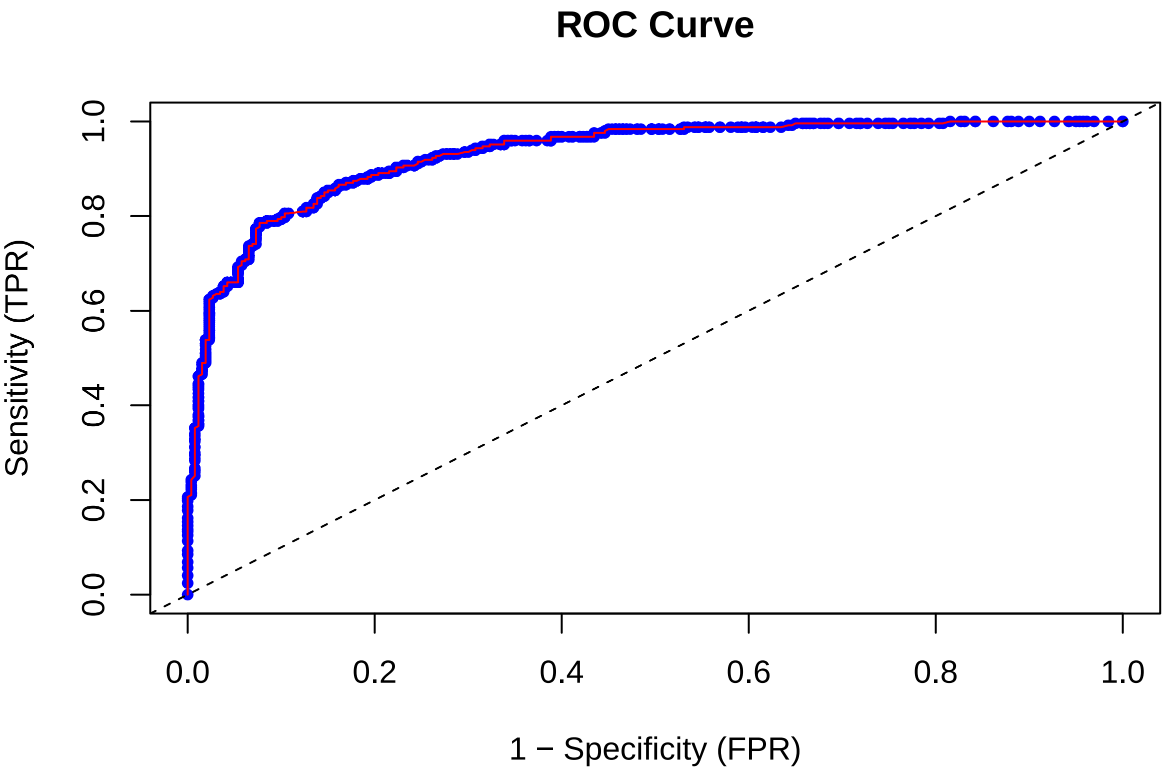
For each threshold, sensitivity (true positive rate, i.e., TP / (TP + FN)) and specificity (TN / (TN + FP)) are calculated. The ROC curve is plotted with specificity on the x-axis and sensitivity on the y-axis. The points are connected by a red line for clarity, and a diagonal reference line is added.

```
# ----------------------------
# 4. Plot the ROC Curve (x: Specificity, y: Sensitivity)
# ----------------------------
plot(1 - roc_data$Specificity, roc_data$Sensitivity,
     xlab = "1 - Specificity (FPR)",
     ylab = "Sensitivity (TPR)",
```

```r
     main = "ROC Curve",
     pch = 20,          # Small filled circles
     col = "blue")      # Points in blue

# Connect the points with a line.
lines(1 - roc_data$Specificity, roc_data$Sensitivity, col = "red")

# Optionally, add a reference line (a random classifier would be near the diagonal of specificity vs. s
abline(0, 1, lty = 2)
```


ROC Curve
Sensitivity (TPR)
1 − Specificity (FPR)