Project 2

Vallesia Pierre Louis

2024-04-22

Import data

```
library(readxl)
library(class)

## Warning: package 'class' was built under R version 4.3.3

library(MASS)
library(pROC)

## Warning: package 'pROC' was built under R version 4.3.3

## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var

dataset <- read_excel("C:/Users/valle/Downloads/STA4504 Project 2
Dataset.xlsx")
str(dataset)

## tibble [4,909 × 12] (S3: tbl_df/tbl/data.frame)
##  $ id               : num [1:4909] 9046 31112 60182 1665 56669 ...
##  $ gender           : chr [1:4909] "Male" "Male" "Female" "Female" ...
##  $ age              : num [1:4909] 67 80 49 79 81 74 69 78 81 61 ...
##  $ hypertension     : num [1:4909] 0 0 0 1 0 1 0 0 1 0 ...
##  $ heart_disease    : num [1:4909] 1 1 0 0 0 1 0 0 0 1 ...
##  $ ever_married     : chr [1:4909] "Yes" "Yes" "Yes" "Yes" ...
##  $ work_type        : chr [1:4909] "Private" "Private" "Private" "Self-
employed" ...
##  $ Residence_type   : chr [1:4909] "Urban" "Rural" "Urban" "Rural" ...
##  $ avg_glucose_level: num [1:4909] 229 106 171 174 186 ...
##  $ bmi              : num [1:4909] 36.6 32.5 34.4 24 29 27.4 22.8 24.2
29.7 36.8 ...
##  $ smoking_status   : chr [1:4909] "formerly smoked" "never smoked"
"smokes" "never smoked" ...
##  $ stroke           : num [1:4909] 1 1 1 1 1 1 1 1 1 1 ...
```

(a) Build a logistic regression model using the following variables: BMI, average glucose level, age, gender, ever married, and work type. State the model. Interpret the coefficient for age in terms of odds.

```
#  logistic regression model
log_model <- glm(stroke ~ bmi + avg_glucose_level + age + gender +
ever_married + work_type, data = dataset, family = binomial)

#summary
summary(log_model)

##
## Call:
## glm(formula = stroke ~ bmi + avg_glucose_level + age + gender +
##     ever_married + work_type, family = binomial, data = dataset)
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -7.759e+00  1.037e+00  -7.483 7.25e-14 ***
## bmi                     6.122e-03  1.173e-02   0.522    0.602
## avg_glucose_level       5.384e-03  1.273e-03   4.231 2.33e-05 ***
## age                     7.640e-02  6.029e-03  12.672  < 2e-16 ***
## genderMale              3.410e-02  1.511e-01   0.226    0.821
## genderOther            -1.139e+01  2.400e+03  -0.005    0.996
## ever_marriedYes        -1.376e-01  2.452e-01  -0.561    0.575
## work_typeGovt_job      -4.960e-01  1.100e+00  -0.451    0.652
## work_typeNever_worked  -1.075e+01  5.094e+02  -0.021    0.983
## work_typePrivate       -3.171e-01  1.087e+00  -0.292    0.770
## work_typeSelf-employed -7.395e-01  1.105e+00  -0.669    0.503
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1381.5  on 4898  degrees of freedom
## AIC: 1403.5
##
## Number of Fisher Scoring iterations: 15
```

The coefficiet for age is 7.640e-02

```
odd_for_age <- exp( 7.640*10^-2 )
cat("The odds ratio for age is ",odd_for_age,"\n")

## The odds ratio for age is  1.079394

percentage <- (odd_for_age -1) *100

cat("FOr every 1 increase in age, the odds of experience a stroke increases
by",percentage,"% while holding other factors constant. ")
```

```
## FOr every 1 increase in age, the odds of experience a stroke increases by
7.939425 % while holding other factors constant.
```

Interpret FOr every 1 increase in age, the odds of experience a stroke increases by 1.07394 while holding other factors constant.

(b) Conduct an overall significance test on the model in part (a). Does at least 1 predictor significantly contribute to the prediction of a stroke?

```
# Wald test for overall significance
wald.test <- summary(log_model)$coefficients[-1, "Pr(>|z|)"]
overall_p_value <- max(wald.test)
tetha <- 0.05

# Check
if (overall_p_value < tetha) {
  cat("p-value <", 0.05, ". The model is overall significant. At least one
predictor significantly contributes to the prediction of a stroke.\n")
} else {
  cat("p-value <", 0.05, ".The model is not overall significant. No predictor
significantly contributes to the prediction of a stroke.\n")
}

## p-value < 0.05 .The model is not overall significant. No predictor
significantly contributes to the prediction of a stroke.
```

p-value < 0.05 .The model is not overall significant. No predictor significantly contributes to the prediction of a stroke

(c) Using the sample proportion of the patients that had a stroke as a cutoff point, find the accuracy, sensitivity, and specificity of the model in part (a).

```
#  probabilities of stroke
predicted_probabilities_of_stroke <- predict(log_model, type = "response")

# cutoff point
cutoff <- mean(dataset$stroke)
predicted_stroke <- ifelse(predicted_probabilities_of_stroke > cutoff, 1, 0)


table <- table(dataset$stroke, predicted_stroke)

#  accuracy
accuracy <- sum(diag(table)) / sum(table)

#  sensitivity
sensitivity <- table[2, 2] / sum(table[2, ])

#  specificity
specificity <- table[1, 1] / sum(table[1, ])
```

```r
# Print results
cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.7331432

cat("Sensitivity:", sensitivity, "\n")

## Sensitivity: 0.7942584

cat("Specificity:", specificity, "\n")

## Specificity: 0.7304255
```

Accuracy: 0.7331432 Sensitivity: 0.7942584 Specificity: 0.7304255

(d) Plot an ROC curve and find the area under the ROC curve. Does it appear that the model in part (a) is better than randomly guessing if a patient had a stroke?

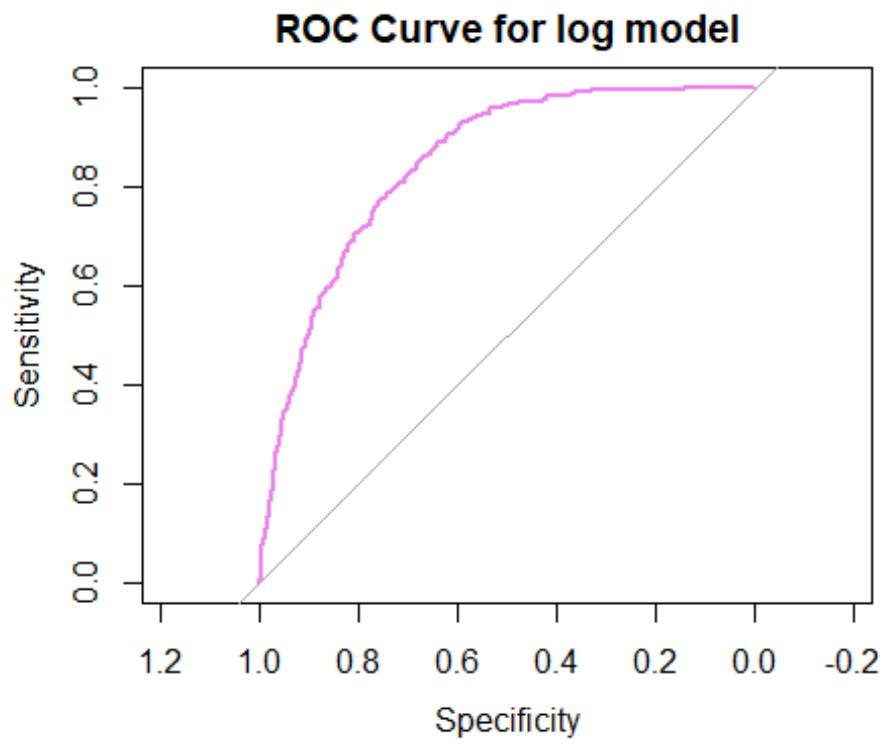
```r
# Prediction probabilities
predictions <- predict(log_model, type = "response")

# Create ROC curve object
roc_curve <- roc(dataset$stroke, predictions)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

#  ROC curve
plot(roc_curve, main = "ROC Curve for log model", col = "violet", lwd = 2)
```




```
#  AUC
auc <- auc(roc_curve)
cat("AUC:", auc, "\n")

## AUC: 0.8458251
```

Yes, it appears that the model in part (a) is better than randomly guessing if a patient had a stroke. The AUC is at 0.8458. Since the AUC is close to 1, It indicates that the model has good performance power. With the AUC, The model is able to distinguish between patients who had a stroke and those who did not with a high amount of accuracy. Therefore, the model is better than randomly guessing.

(e) Using the model in part (a) as the full model, conduct backwards selection using AIC as the criterion. State the resultant model.

```
# Backward selection
backward_selection_model <- step(log_model, direction = "backward", trace =
0)

summary(backward_selection_model)

##
## Call:
## glm(formula = stroke ~ avg_glucose_level + age, family = binomial,
##     data = dataset)
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)       -7.863852   0.383221 -20.520  < 2e-16 ***
## avg_glucose_level  0.005597   0.001229   4.555 5.23e-06 ***
## age                0.071836   0.005423  13.246  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1728.4  on 4908  degrees of freedom
## Residual deviance: 1387.9  on 4906  degrees of freedom
## AIC: 1393.9
##
## Number of Fisher Scoring iterations: 7
```

The resultant model is -7.863852 +0.005597(avg_glucose_level) + 0.071836(age)

(f) Again, using the sample proportion of the patients that had a stroke as a cutoff point, find the accuracy, sensitivity, and specificity of the model in part (e).

```
#sample proportion
cutoff<- mean(dataset$stroke)

dataset$predicted_stroke_e <-ifelse(predict(backward_selection_model,type =
"response") > cutoff, 1, 0)

#  accuracy
accuracy_two <- mean(dataset$stroke == dataset$predicted_stroke_e)

#  sensitivity
true_positive_e <- sum(dataset$stroke == 1 & data$predicted_stroke_e == 1)

## Error in data$predicted_stroke_e: object of type 'closure' is not
subsettable

actual_positive <- sum(dataset$stroke == 1)
sensitivity_two <- true_positive_e / actual_positive

## Error in eval(expr, envir, enclos): object 'true_positive_e' not found

#  specificity
true_negative_e <- sum(dataset$stroke == 0 & dataset$predicted_stroke_e == 0)
actual_negative <- sum(dataset$stroke == 0)
specificity_two <- true_negative_e / actual_negative

# Print the results
cat("Accuracy (Backwards Model):", accuracy_two, "\n")

## Accuracy (Backwards Model): 0.7331432

cat("Sensitivity (Backwards Model):", sensitivity_two, "\n")

## Error in eval(expr, envir, enclos): object 'sensitivity_two' not found
```

```r
cat("Specificity (Backwards Model):", specificity_two, "\n")
```

```
## Specificity (Backwards Model): 0.73
```

Accuracy (Backwards Model): 0.7331432 Sensitivity (Backwards Model): 0.8038278 Specificity (Backwards Model): 0.73

(g) [10 pts] Plot an ROC curve and find the area under the ROC curve. Does it appear that the model in part (e) is better than randomly guessing if a patient had a stroke?

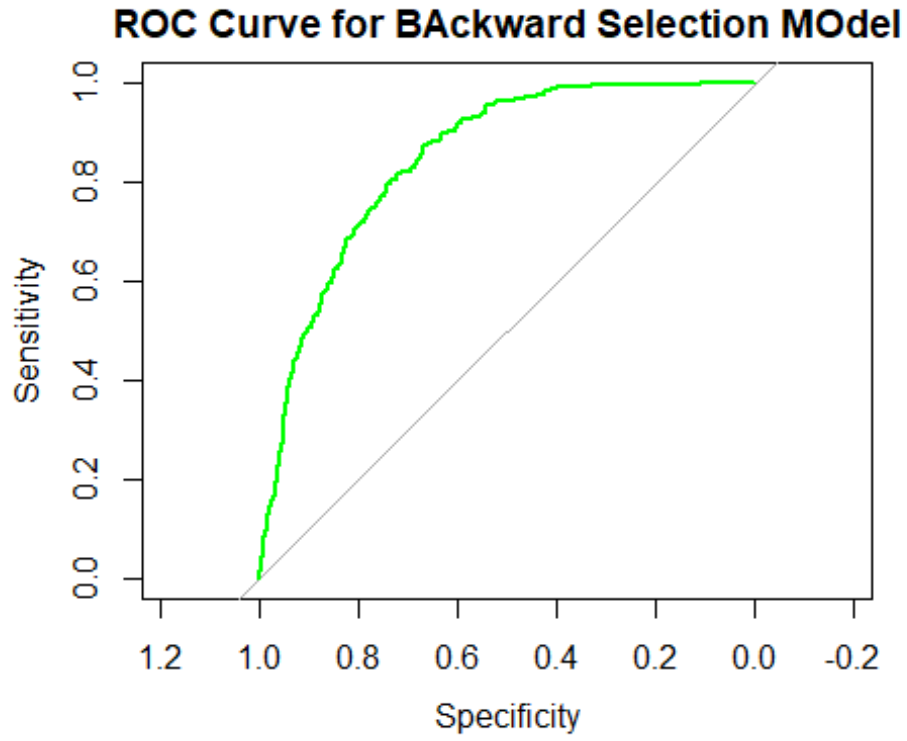
```r
# Prediction probabilities for model e
predictions_e <- predict(backward_selection_model, type = "response")

# Create a ROC curve object for model e
roc_obj_e <- roc(dataset$stroke, predictions_e)
```

```
## Setting levels: control = 0, case = 1
```

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

```r
# Plot ROC curve for model e
plot(roc_obj_e, main = "ROC Curve for BAckward Selection MOdel", col =
"green", lwd = 2)
```




```r
# Calculate AUC for model e
auc_e <- auc(roc_obj_e)
cat("AUC (Model e):", auc_e, "\n")
```

```
## AUC (Model e): 0.8447903
```

(h) Compare the models from part (a) and (e). Which model would you suggest we use for prediction? Make sure to include some metrics and rationale on which model you would choose.

Both models have the same accuracy,specificity and AUC. I chose model e, Backward Selection model. Model e had a higher sensitivity at 0.804. MOdel e also has a lower complexity.

Bonus Questions: (i) Use K-Nearest Neighbors (with 5 neighbors) using all the continuous predictors listed in part (a) (age, average glucose level, and BMI). Find the accuracy, sensitivity, and specificity of the model.

```
continuous_predictors <- dataset[, c("age", "avg_glucose_level", "bmi")]

# Split the data into predictors and outcome
X <- as.matrix(continuous_predictors)
y <- as.factor(dataset$stroke)

# Fit KNN model with 5 neighbors
knn_model <- knn(train = X, test = X, cl = y, k = 5)
c_matrix_one <-(table(knn_model,y))
print(c_matrix_one)
```

```
##          y
## knn_model    0    1
##         0 4689  191
##         1   11   18
```

```
#Find the accuracy and other metrics

#accuracy
accuracy <- (18+4689) / sum(c_matrix_one)
cat("Accuracy:", accuracy, "\n")
```

```
## Accuracy: 0.9588511
```

```
#sensitivity
sensitivity <- 18 / sum(18+11)
cat("Sensitivity:", sensitivity, "\n")
```

```
## Sensitivity: 0.6206897
```

```
#Specificity
specificity <- 4689 / sum(4646+191)
cat("Specificity:", specificity, "\n")
```

```
## Specificity: 0.9694025
```

Accuracy: 0.9588511 Sensitivity: 0.6206897 Specificity: 0.9694025

(j) Use Linear Discriminant Analysis using all the predictors listed in part (a). Find the accuracy, sensitivity, and specificity of the model.

```
library(MASS)

#Fit the lda model
lda.fit <- lda(stroke ~ ., data=dataset)

#Find the accuracy and other metrics
lda.pred = predict(lda.fit, dataset)
lda.class=lda.pred$class

c_matrix<-(table(lda.class, dataset$stroke))
print(c_matrix)

##
## lda.class    0    1
##         0 4649  188
##         1   51   21

#accuracy
accuracy <- (23+4646) / sum(c_matrix)
cat("Accuracy:", accuracy, "\n")

## Accuracy: 0.9511102

#sensitivity
sensitivity <- 23 / sum(23+54)
cat("Sensitivity:", sensitivity, "\n")

## Sensitivity: 0.2987013

#Specificity
specificity <- 4646 / sum(4646+186)
cat("Specificity:", specificity, "\n")

## Specificity: 0.9615066
```

Accuracy: 0.9511102 Sensitivity: 0.2987013 Specificity: 0.9615066

(k) Which model would you suggest using among the logistic regression full model, logistic regression reduced model, KNN, and LDA.

The best model is the KNN MOdel.