Homework Assignment 2 PSTAT 131

Quinlan Wilson and Jack Guo (both 131)

November 06, 2025

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
##
## Attaching package: 'MASS'
##
##
## The following object is masked from 'package:dplyr':
##
##     select
```

Linear Regression

(1.)

```
glimpse(Auto)
```

```
## Rows: 392
## Columns: 9
## $ mpg          <dbl> 18, 15, 18, 16, 17, 15, 14, 14, 14, 15, 15, 14, 15, 14, 2~
## $ cylinders    <dbl> 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 6, 6, 6, 4, ~
## $ displacement <dbl> 307, 350, 318, 304, 302, 429, 454, 440, 455, 390, 383, 34~
## $ horsepower   <dbl> 130, 165, 150, 150, 140, 198, 220, 215, 225, 190, 170, 16~
## $ weight       <dbl> 3504, 3693, 3436, 3433, 3449, 4341, 4354, 4312, 4425, 385~
## $ acceleration <dbl> 12.0, 11.5, 11.0, 12.0, 10.5, 10.0, 9.0, 8.5, 10.0, 8.5, ~
## $ year         <dbl> 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 70, 7~
## $ origin       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 3, ~
## $ name         <fct> chevrolet chevelle malibu, buick skylark 320, plymouth sa~
```

```
Auto_clean <- Auto %>%
  dplyr::select(-contains("name")) %>%
  mutate(origin = as.factor(origin))
```

```r
lmod <- lm(mpg ~ ., data = Auto_clean)
summary(lmod)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = Auto_clean)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.009 -2.078 -0.098  1.986 13.361
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.80e+01   4.68e+00   -3.84  0.00014 ***
## cylinders    -4.90e-01   3.21e-01   -1.52  0.12821
## displacement  2.40e-02   7.65e-03    3.13  0.00186 **
## horsepower   -1.82e-02   1.37e-02   -1.33  0.18549
## weight       -6.71e-03   6.55e-04  -10.24  < 2e-16 ***
## acceleration  7.91e-02   9.82e-02    0.81  0.42110
## year          7.77e-01   5.18e-02   15.01  < 2e-16 ***
## origin2       2.63e+00   5.66e-01    4.64  4.7e-06 ***
## origin3       2.85e+00   5.53e-01    5.16  3.9e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.31 on 383 degrees of freedom
## Multiple R-squared:  0.824,  Adjusted R-squared:  0.821
## F-statistic:  224 on 8 and 383 DF,  p-value: <2e-16
```

With a 0.01 threshold we can reject the null hypothesis that there is no linear association between mpg and displacement, weight, year, and origin. While failing to reject that cylinders, horsepower, and acceleration have no linear association with mpg.

(2.)

```r
training_MSE <- mean((Auto$mpg - predict(lmod, Auto_clean))^2)
```

The training MSE is 10.6821 and we cannot calculate the test MSE as the training set is the whole dataset so we have nothing to compare with.

(3.)

```r
car <- data.frame(
  cylinders = 4,
  displacement = 133,
  horsepower = 117,
  weight = 3250,
  acceleration = 29,
```

```
  year = 97,
  origin = factor(2)
)


car_pre_mpg <- predict(lmod, newdata = car)
```

The gas mileage predicted for an European car with 4 cylinders, displacement 133, horsepower of 117, weight of 3250, acceleration of 29, built in the year 1997 is 39.6351.

(4.)

```
coef(lmod)
```

```
##  (Intercept)    cylinders displacement   horsepower       weight acceleration
##    -17.95460     -0.48971      0.02398     -0.01818     -0.00671      0.07910
##         year      origin2      origin3
##      0.77703      2.63000      2.85323
```

The difference in mpg between a Japanese car and the mpg of an American car is 2.85323 and the difference between a European car and an American car is 2.63000.

(5.)

The change in mpg associated with a 30 unit increase in horsepower is -0.5455.

```
algae <- read.table("algaeBloom.txt", col.names=
                      c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4',
                        'oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),
                    na = "XXXXXXX")
```

```
algae.transformed <- algae %>% mutate_at(vars(4:11), funs(log(.)))
algae.transformed <- algae.transformed %>%
mutate_at(vars(4:11),funs(ifelse(is.na(.),median(.,na.rm=TRUE),.)))
# a1 == 0 means low
algae.transformed <- algae.transformed %>% mutate(a1 = factor(as.integer(a1 > 5), levels = c(0, 1)))
```

```
calc_error_rate <- function(predicted.value, true.value) {
  return(mean(true.value != predicted.value))
}
```

```
set.seed(1)
test.indices = sample(1:nrow(algae.transformed), 50)
algae.train=algae.transformed[-test.indices,]
algae.test=algae.transformed[test.indices,]
```

Algae Classification using Logistic regression

```
library(tidyverse)
library(ISLR)
library(ROCR)

algae <- read_table2("algaeBloom.txt", col_names=
c('season','size','speed','mxPH','mnO2','Cl','NO3','NH4',
'oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),
na="XXXXXXX")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   season = col_character(),
##   size = col_character(),
##   speed = col_character(),
##   mxPH = col_double(),
##   mnO2 = col_double(),
##   Cl = col_double(),
##   NO3 = col_double(),
##   NH4 = col_double(),
##   oPO4 = col_double(),
##   PO4 = col_double(),
##   Chla = col_double(),
##   a1 = col_double(),
##   a2 = col_double(),
##   a3 = col_double(),
##   a4 = col_double(),
##   a5 = col_double(),
##   a6 = col_double(),
##   a7 = col_double()
## )
```

```
## Take log of vars
## Fill in missing
## categorize into high/low based on a1
algae.transformed <- algae %>% mutate_at(vars(4:11), funs(log(.)))
algae.transformed <- algae.transformed %>%
mutate_at(vars(4:11),funs(ifelse(is.na(.),median(.,na.rm=TRUE),.)))
# a1 == 0 means low
algae.transformed <- algae.transformed %>% mutate(a1 = factor(as.integer(a1 > 5), levels = c(0, 1)))

## Misclassification error rate function
calc_error_rate <- function(predicted.value, true.value){
return(mean(true.value != predicted.value))
}

## Training/test sets
set.seed(1)
test.indices = sample(1:nrow(algae.transformed), 50)
algae.train=algae.transformed[-test.indices,]
algae.test=algae.transformed[test.indices,]
```

(1.)

$z = \ln(\frac{p}{1-p})$

$e^z = \frac{p}{1-p}$

$e^z - pe^z - p = 0$

$p(1 - e^z) = e^z$

$p = \frac{e^z}{1-e^z}$

(2.)

Substitute $z = \beta_0 + \beta_1 x_1$ into odds.

log odds $= log(\frac{p}{1-p}) = z$

$odds = \frac{p}{1-p} = e^z$

$odds = e^{\beta_0 + \beta_1 x_1}$

$odds = e^{\beta_0 + \beta_1 (x_1 + 2)}$

$odds = e^{\beta_0 + \beta_1 x_1} * e^{2\beta_1}$

The odds increase by a factor of $e^{2\beta_1}$.

If β_1 is negative, p approaches 0 as x_1 approaches ∞. p approaches 1 as x_1 approaches $-\infty$.

(3.)

```
model <- glm(a1 ~ . - a2:a7, data=algae.transformed, family="binomial")
predicted <- data.frame(predicted_val=predict(model, algae.test, type="response"))
predicted_classified <- predicted %>% mutate(
  a1=as.integer(predicted_val>.5)
)
predicted_classified ## predictions of test data
```

```
##    predicted_val a1
## 1        0.99913  1
## 2        0.09725  0
## 3        0.10122  0
## 4        0.15638  0
## 5        0.04156  0
## 6        0.97044  1
## 7        0.51267  1
## 8        0.85502  1
## 9        0.33318  0
## 10       0.16492  0
## 11       0.74999  1
## 12       0.92336  1
## 13       0.10626  0
## 14       0.72006  1
## 15       0.20568  0
## 16       0.55254  1
```

```
## 17       0.99897  1
## 18       0.53615  1
## 19       0.42116  0
## 20       0.63624  1
## 21       0.97285  1
## 22       0.06838  0
## 23       0.26847  0
## 24       0.10747  0
## 25       0.07212  0
## 26       0.30607  0
## 27       0.38164  0
## 28       0.28898  0
## 29       0.47288  0
## 30       0.48040  0
## 31       0.34884  0
## 32       0.17341  0
## 33       0.96827  1
## 34       0.65309  1
## 35       0.07975  0
## 36       0.14019  0
## 37       0.01686  0
## 38       0.03369  0
## 39       0.45012  0
## 40       0.01283  0
## 41       0.03944  0
## 42       0.22581  0
## 43       0.96846  1
## 44       0.09860  0
## 45       0.03322  0
## 46       0.12711  0
## 47       0.14369  0
## 48       0.08707  0
## 49       0.99277  1
## 50       0.53527  1
```

```
## error rates
calc_error_rate(predicted_classified$a1, algae.train$a1)
```

```
## [1] 0.5733
```

```
calc_error_rate(predicted_classified$a1, algae.test$a1)
```

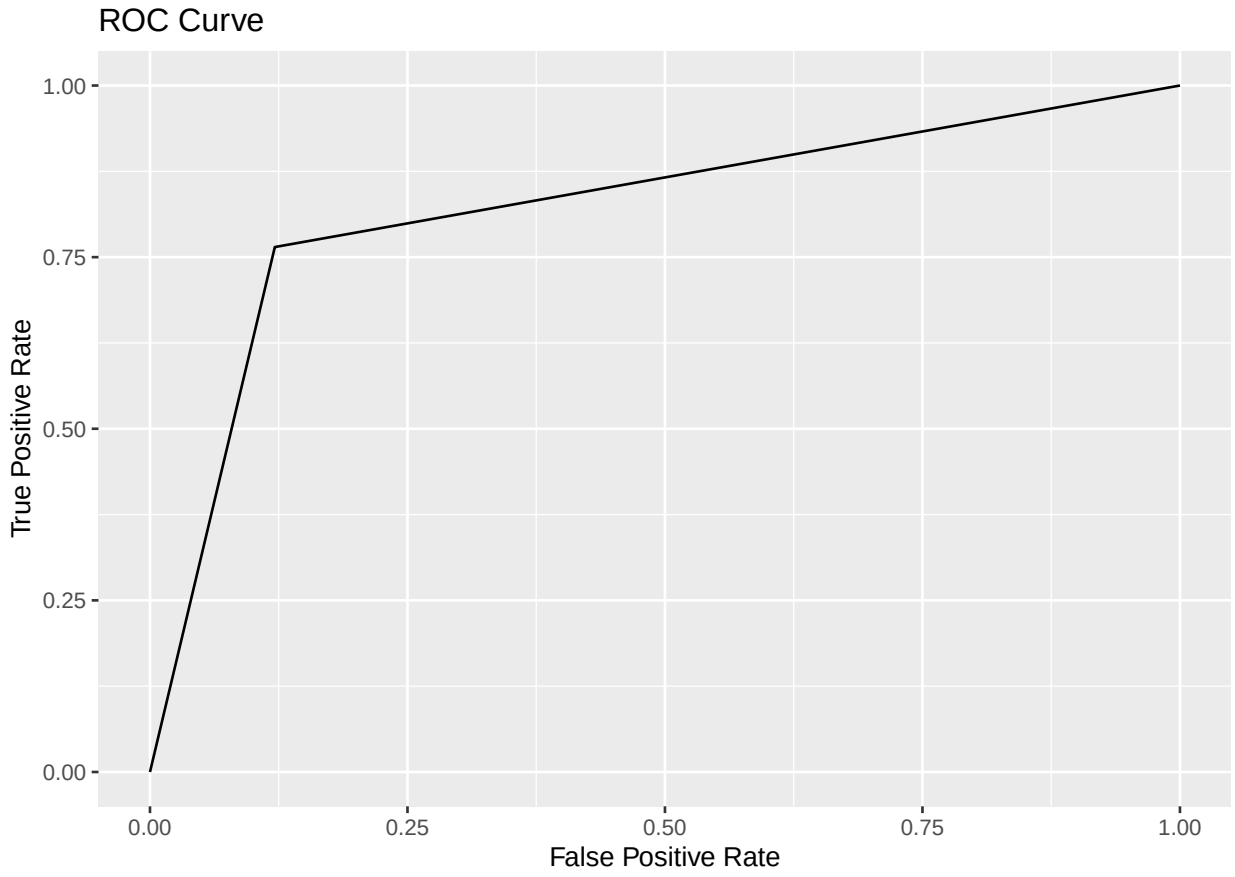
```
## [1] 0.16
```

The training error is 0.573333 and the test error is 0.16.

(4.)

```
pred <- prediction(predicted_classified$a1, algae.test$a1)
perf <- performance(pred, "tpr", "fpr")
```

```r
roc_data <- data.frame(
  fpr = unlist(perf@x.values),
  tpr = unlist(perf@y.values)
)
ggplot(roc_data, aes(x=fpr, y=tpr)) +
  geom_line() +
  labs(
    title="ROC Curve",
    x="False Positive Rate",
    y="True Positive Rate"
  )
```



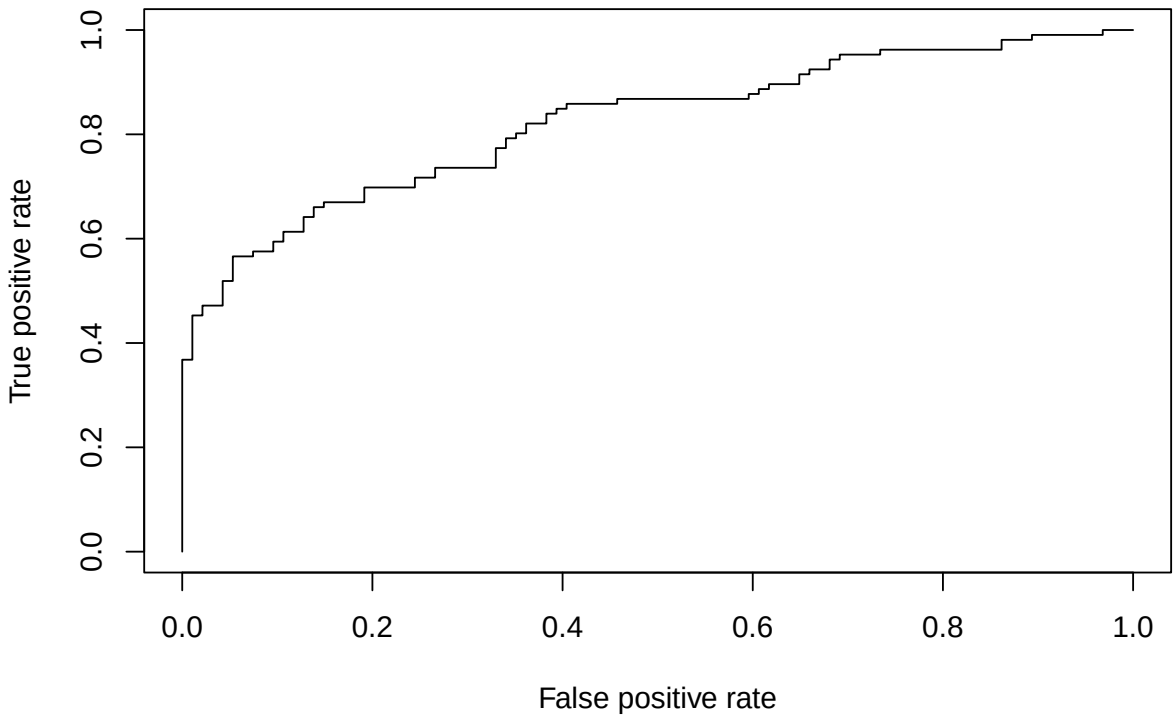

```r
auc <- performance(pred,"auc")
auc_value <- auc@y.values[[1]]
print(paste("AUC =", auc_value))
```

```
## [1] "AUC = 0.82174688057041"
```

Algae Classification using Discriminant Analysis

(1.)

```
lda_mod <- lda(a1 ~ ., data = algae.transformed, CV = T)


lda_class <- lda_mod$class

lda_error <- calc_error_rate(lda_class, algae.train$a1)
lda_post <- lda_mod$posterior

if (ncol(lda_post) == 1) {
  lda_post <- cbind("0" = 1 - lda_post, "1" = lda_post)
}

lda_probs <- lda_post[, "1"]

pred <- prediction(lda_mod$posterior[, "1"], algae.transformed$a1)
perf <- performance(pred, 'tpr', 'fpr')

plot(perf)
```

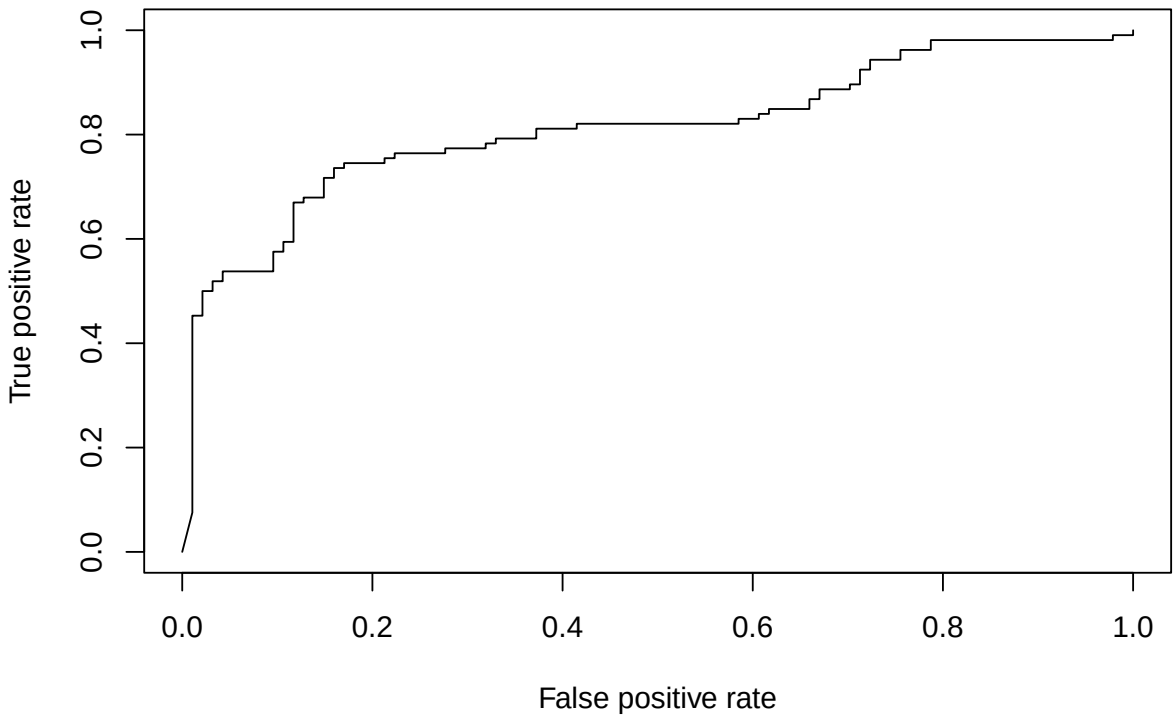


```r
pred_lda <- prediction(lda_probs, algae.transformed$a1)
performance(pred_lda, "auc")@y.values[[1]]
```

```
## [1] 0.8232
```

(2.)

```r
qda_mod <- qda(a1 ~ ., data = algae.transformed, CV = T)
qda_class <- qda_mod$class
qda_post <- qda_mod$posterior

if (ncol(qda_post) == 1) qda_post <- cbind("0" = 1 - qda_post, "1" = qda_post)

qda_probs <- qda_post[, "1"]

pred_qda <- prediction(qda_probs, algae.transformed$a1)
perf_qda <- performance(pred_qda, "tpr", "fpr")
auc_qda <- performance(pred_qda, "auc")@y.values[[1]]

plot(perf_qda)
```



```
auc_qda
```

```
## [1] 0.8131
```

LDA has a higher area under the ROC curve between the two models meaning it performs better at separating the two classes. LDA has the advantage of lower variance but in the case of different contrivances more bias is introduced.

Fundamentals of the bootstrap

(1.)

$$(1 - \frac{1}{n})^n$$

(2.)

```
(1-1/1000)^1000
```

```
## [1] 0.3677
```

(3.)

```
n <- 1000

sample <- sample(1:n, size = n, replace = T)

missing <- n - length(unique(sample))

ratio <- missing / n
ratio
```

```
## [1] 0.371
```

Cross-validation estimate of test error

(1.)

```
dat = subset(Smarket, select = -c(Year,Today))
dat$Direction = ifelse(dat$Direction == "Up", 1,0)

set.seed(123)
test.indices = sample(1:nrow(dat), 1250/2)
train=dat[-test.indices,]
test=dat[test.indices,]
```

```r
model <- glm(Direction ~ ., data=train, family="binomial")
pred <- data.frame(predicted_val=predict(model, test, type="response"))
pred_classified <- pred %>% mutate(
  Direction=as.integer(predicted_val>.5)
)
calc_error_rate(pred_classified$Direction, test$Direction)
```

```
## [1] 0.4784
```

(2.)

```r
do.chunk <- function(chunkid, folddef, dat, ...){
# Get training index
train = (folddef!=chunkid)
# Get training set and validation set
dat.train = dat[train, ]
dat.val = dat[-train, ]
# Train logistic regression model on training data
fit.train = glm(Direction ~ ., family = binomial, data = dat.train)
# get predicted value on the validation set
pred.val = predict(fit.train, newdata = dat.val, type = "response")
pred.val = ifelse(pred.val > .5, 1,0)
data.frame(fold = chunkid,
val.error = mean(pred.val != dat.val$Direction))
}

folddef <- sample(rep(1:10, length.out = nrow(dat)))

results <- data.frame()
for (k in 1:10) {
  results <- rbind(results, do.chunk(k, folddef, dat))
}
mean(results$val.error)
```

```
## [1] 0.4758
```

The average validation error for the 10-fold cross-validation was 0.4698959.