Daniel Rance
49144736
dra124

STAT462: Data Mining

Assignment 2

1. x1 = hours studied each week, x2 = number of classes attended and,

$$Y = \begin{cases} 1 \; if \; the \; student \; received \; a \; GPA \; value \geq 7 \; in \; STAT462 \\ 0 \; otherwise \end{cases}$$

After fitting a logistic regression model, we receive our estimated coefficients:

β0(intercept) = -16
β1(hours studied) = 1.4
β2(classes attended) = 0.3

a) Estimate the probability of a student getting a GPA value ≥7 (Y = 1) in Stat462 if they study for 5 hours per week (x1 = 5) and attend all 36 classes (x2 = 36).

The logistic function takes the form:

$$\Pr(Y = 1|X = x) = p(x) = \frac{\exp(\beta 0 + \beta 1 x1 + \beta 2 x2)}{1 + \exp(\beta 0 + \beta 1 x1 + \beta 2 x2)}$$

So, in the above example, our equation is:

$$p(x) = \frac{\exp(-16 + 5(1.4) + 0.3(36)}{1 + \exp(-16 + 5(1.4) + 0.3(36)}$$

Which gives us a result of 0.858, meaning there is an approximately 86% chance of a student that studies for 5 hours per week and attends all 36 lectures to receive a GPA ≥ 7 in STAT462.

Daniel Rance
49144736
dra124

b) If a student attends 18 classes (x2 = 18), how many hours do they need to study per week (x1 =?) to have a 50% chance(p(x) = 0.5) of getting a GPA ≥ 7 in STAT462?

We can rearrange the logistic function to obtain log odds,

$$p(x) = \frac{\exp(\beta 0 + \beta 1 x1 + \beta 2 x2)}{1 + \exp(\beta 0 + \beta 1 x1 + \beta 2 x2)}$$

Becomes:

$$log\left(\frac{p(x)}{1 - p(x)}\right) = \beta 0 + B1 x1 + \beta 2 x2$$

We are trying to discover x1, i.e., how many hours of study is required per week to have a 50 percent chance (p(x) = 0.5), given that the student has attended 18 classes (x2). We can rearrange the function again to find x1:

$$x1 = \frac{log\left(\frac{p(x)}{1 - p(x)}\right) - \beta 0 - \beta 2 x2}{\beta 1}$$

Using our own figures,

$$log\left(\frac{0.5}{1 - 0.5}\right) = -16 + 1.4(x1) + 0.3(18)$$

$$log(1) = -16 + 5.4 + 1.4(x1)$$

$$0 = -16 + 5.4 + 1.4(x1)$$

$$10.6 = 1.4x1$$

$$x1 = \frac{10.6}{1.4} = \mathbf{7.57}$$

Meaning a student would need to study 7.57 hours per week to have a 50% chance of getting a GPA value ≥ 7 in STAT462.

Daniel Rance
49144736
dra124

2. **Using the BankTrain.csv & BankTest.csv data sets, fit a logistic regression to predict the probability of a banknote being forged. The response variable is y, where y = 1 denotes a forged banknote and y = 0 denotes a genuine banknote. We will be using x1 and x3 to fit our model.**

a) **Perform multiple logistic regression using the training data. Comment on the model obtained.**

We can attach our training data set and perform multiple logistic regression using the following R code:

```
library(ISLR)

BankTrain=read.csv("BankTrain.csv",header=T,na.strings="?")
BankTest=read.csv("BankTest.csv",header =T,na.strings="?")
names(BankTrain)

attach(BankTrain)
glm.fit=glm(y~x1+x3, data=BankTrain, family=binomial)
summary(glm.fit)

##
## Call:
## glm(formula = y ~ x1 + x3, family = binomial, data = BankTrain)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.83187  -0.28343  -0.06417   0.50032   1.99366
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22041    0.11206   1.967   0.0492 *
## x1          -1.31489    0.08822 -14.905  < 2e-16 ***
## x3          -0.21738    0.02880  -7.548 4.42e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1322.01  on 959  degrees of freedom
## Residual deviance:  572.07  on 957  degrees of freedom
## AIC: 578.07
##
## Number of Fisher Scoring iterations: 6
```

Daniel Rance
49144736
dra124

Here we can see that the p values for x1 (the variance of a Wavelet Transformed image) and x3 (the kurtosis of a Wavelet Transformed Image) both very small ($p < 0.05$), suggesting a significant statistical relationship between both of our predictors and the response variable y(forged vs non-forged). The intercept is also weakly significant.

b) Suppose we classify observations using:

$$Y = \begin{cases} forged\ banknote & if\ \Pr(Y = 1|X = x) > 0 \\ genuine\ banknote & otherwise \end{cases}$$

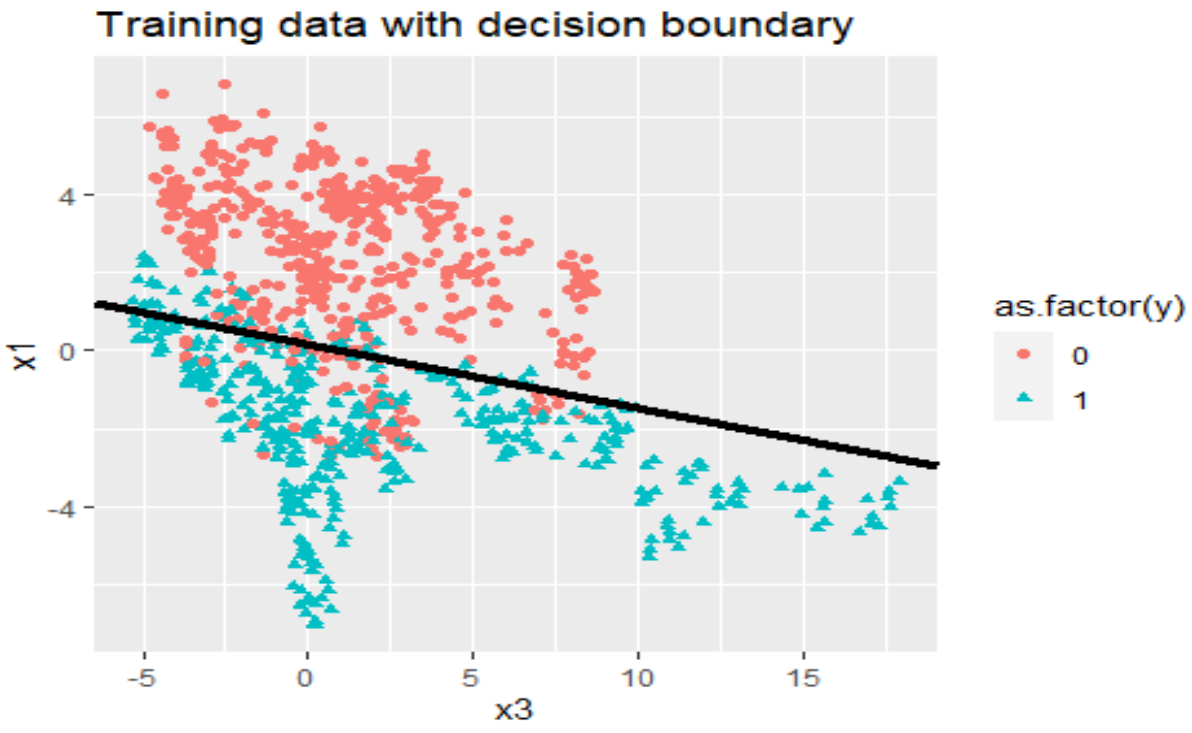
i. Plot the training data(using a different symbol for each class) and the decision boundary for θ = 0.5 on the same figure

```
library(ggplot2)

b0 = glm.fit$coefficients[1]
b1 = glm.fit$coefficients[2]
b3 = glm.fit$coefficients[3]

#decision boundary for theta 0.5
x3 = seq(from = -7, to = 18, by = 0.1)
x1 = -(b0/b1) - (b3/b1) * x3

ggplot(BankTrain, aes(x=x3, y=x1, shape=as.factor(y), color=as
.factor(y))) +
  geom_point() +
  geom_abline(intercept = -(b0/b1), slope = -(b3/b1), color =
'black', size=1.5) +
  ggtitle("Training data with decision boundary")
```



ii. Using θ = 0.5, computer the confusion matrix for the testing set and comment on your output.

Confusion Matrix for θ = 0.5

```
glm.probs = predict(glm.fit, newdata = BankTest, type='respons
e')
glm.pred=rep(0,412) #dimensions of the test data
glm.pred[glm.probs>.5]=1 #anything with a prob greater than .5
is considered a forgery
table(glm.pred, BankTest$y)

##
## glm.pred   0   1
##        0 204  24
##        1  32 152
```

Here the horizontal axis represents the true state of the banknote in the testing set (Not forged = 0, forged = 1), while the vertical axis represents the predicted state of the banknote in the testing set, using our logistic regression model and θ = 0.5.

The **testing error** is (24+32)/412 or 13.59%. The **null classifier** (assign all unclassified observations to the majority class) has a testing error of (24+152)/412 or 42.72%. Our model is significantly better at predicting bank note forgeries than the null classifier.

For **true non-forged bank notes**, we have an error rate of 32/236 or **13.6%.**

For **true forged bank notes,** we have an error rate of 24/176 or **13.6%.**

In this example our **Precision (TP/P*)** is (152/184) **82.6%**, our **Specificity (True Negative Rate – TN/N)** is (204/236) **86.4%** and our **Sensitivity (True Positive Rate – TP/P)** is (152/176) **86.4%.**

Our model has approximately the same prediction power when predicting true forged bank notes and true non-forged bank notes, it also has approximately identical True Negative and True Positive Rates (Specificity & Sensitivity). This is most likely due to the similar sample sizes present in our testing data. As a bank or government institution interested in the legitimacy of circulated currency, we may decide that we are more interested in increasing our True Positive Rate/Sensitivity to identify as many forged bank notes as possible, even if that means

sacrificing our True Negative Rate/Specificity somewhat. To do so, we can adjust our **θ**, or our **threshold.**

iii. **Compute confusion matrices for the testing set using θ = 0.3 and θ = 0.6. Comment on your output. Describe a situation where the θ = 0.3 model may be the preferred model.**

Confusion Matrix for θ = 0.3

```
glm.probs = predict(glm.fit, newdata = BankTest, type='respons
e')
glm.pred=rep(0,412) #dimensions of the test data
glm.pred[glm.probs>.3]=1 #anything with a prob greater than .3
is considered a forgery
table(glm.pred, BankTest$y)

##
## glm.pred   0   1
##        0 183   5
##        1  53 171
```

Using a θ = 0.3, the **Testing Error** is (53+5)/412 or **14.1%.** This is a slight increase from our θ = 0.5 model Testing Error of 13.59% but still significantly better than our **null classifier** of (5+171)/412 or **42.7%.**

For **true non-forged bank notes,** we have an error rate of 53/236 or **22.5%.** This, again, is a decrease from our θ = 0.5 model.

For **true forged bank notes,** we have an error rate of 5/176 or **2.8%.** This is a *significant* improvement from our θ = 0.5 model.

In this example our **Precision (TP/P*)** is (171/224) **76.3%**, our **Specificity (True Negative Rate – TN/N)** is (183/236) **77.5%** and our **Sensitivity (True Positive Rate – TP/P)** is (171/176) **97.1%.**

Using θ = 0.3 for our model, we have slightly increased our overall testing error (13.59% -> 14.1%). We have decreased our ability to predict non-forged bank notes (error rate went from 13.6% -> 22.5%) whilst simultaneously significantly increasing our ability to predict forged bank notes (error rate went from 13.6% -> 2.8%). We have also seen decreases in both our Precision and Specificity. Where we see a significant improvement, however, is in our Sensitivity. This has had a significant improvement (86.4% -> 97.1%).

Daniel Rance
49144736
dra124

Confusion Matrix for θ = 0.6

```
glm.probs = predict(glm.fit, newdata = BankTest, type='respons
e')
glm.pred=rep(0,412) #dimensions of the test data
glm.pred[glm.probs>.6]=1 #anything with a prob greater than .6
is considered a forgery
table(glm.pred, BankTest$y)

##
## glm.pred   0   1
##        0 210  35
##        1  26 141
```

Using a θ = 0.6, the **Testing Error** is (26+35)/412 or **14.8%.** Again, this is a worsening from the testing error in our θ = 0.5 & θ = 0.3 models but still significantly better than our **null classifier** of (35+141)/412 of **42.7%.**

For **true non-forged bank notes,** we have an error rate of 26/236 or **11%.** This is an improvement on both our θ = 0.5 & θ = 0.3 models.

For **true forged bank notes,** we have an error rate of 35/176 or **19.9%.** This is a decrease from both our θ = 0.5 & θ = 0.3 models.

In this example our **Precision (TP/P*)** is (141/167) **84.4%**, our **Specificity (True Negative Rate – TN/N)** is (210/236) **88.9%** and our **Sensitivity (True Positive Rate – TP/P)** is (141/176) **80.1%.**

Using θ = 0.6 for our model, we have slightly increased our overall testing error, compared to θ = 0.5 (13.59% -> 14.8%). We have increased our ability to predict non-forged bank notes (error rate went from 13.6% -> 11%) whilst simultaneously decreasing our ability to predict forged bank notes (error rate went from 13.6% -> 19.9%). We have also seen increases in both our Precision and Specificity. However, we have seen a decrease in our Sensitivity (86.4% -> 80.1%).

Like I described in section ii, I believe there is a situation where a bank/other financial institution would be willing to sacrifice other areas of model accuracy to ensure the greatest *level of sensitivity* possible i.e., catching as many forged banknotes as possible. This may be due to a cost differential between a low-cost false positive and a higher cost false negative (keeping forged notes in circulation would cost the bank

more than accidentally removing real notes from circulation). In this scenario, I believe the bank/other financial institution would prefer θ = 0.3 for their model as it has the greatest level of sensitivity/true positive rate (97.1%) whilst its overall error rate still performs better than the null classifier.

3. Fit LDA & QDA models to the training set from question 2 of this assignment.

a) Fit an LDA model to predict the probability of a banknote being forged using the predictors x1 & x3. Compute the confusion matrix for the testing set from question 2.

#Installing MASS library and fitting an LDA model to the **training set**

```
library(MASS)
BankTrain.lda <- lda(y ~ x1 + x3, data = BankTrain)
BankTrain.lda
```

```
## Call:
## lda(y ~ x1 + x3, data = BankTrain)
##
## Prior probabilities of groups:
##         0         1
## 0.5479167 0.4520833
##
## Group means:
##          x1       x3
## 0  2.322977 0.938296
## 1 -1.870594 2.114927
##
## Coefficients of linear discriminants:
##            LD1
## x1 -0.55425154
## x3 -0.07209638
```

#Examining the names inside our model

```
lda.pred=predict(BankTrain.lda, BankTest)
names(lda.pred)
```

```
## [1] "class"     "posterior" "x"
```

#Calcuating confusion matrix for our **testing set**

```
lda.class = lda.pred$class
table(lda.class, BankTest$y)
```

```
##
## lda.class   0   1
##         0 203  22
##         1  33 154
```

#Assessing Accuracy

```
mean(lda.class==BankTest$y)
```

```
## [1] 0.8665049
```

b) Repeat part (a) using QDA.

#Installing MASS library and fitting an QDA model to the training set

```
library(MASS)
BankTrain.qda = qda.fit=qda(y~x1+x3,data=BankTrain)
BankTrain.qda
```

```
## Call:
## qda(y ~ x1 + x3, data = BankTrain)
##
## Prior probabilities of groups:
##         0         1
## 0.5479167 0.4520833
##
## Group means:
##          x1       x3
## 0  2.322977 0.938296
## 1 -1.870594 2.114927
```

#Examining the names inside our model

```
qda.pred=predict(BankTrain.qda, BankTest)
names(qda.pred)
```

```
## [1] "class"     "posterior"
```

#Calcuating confusion matrix for our testing set

```
qda.class = qda.pred$class
table(qda.class, BankTest$y)
```

```
##
## qda.class   0   1
##         0 208  18
##         1  28 158
```

#Assessing Accuracy

```
mean(qda.class==BankTest$y)
```

```
## [1] 0.8883495
```

Daniel Rance
49144736
dra124

c) **Comment on your results from parts (a) and (b). Compare these methods with the logistic regression model (using θ = 0.5) from question 2. Which method would you recommend for this problem and why?**

Considering the earlier assumption made regarding high-cost false negatives vs low-cost false positives (much more concerned with correctly identifying forged banknotes), I will be focusing on two key measures of model accuracy – **Testing Error** & **Sensitivity.**

The **testing error** for the **LDA** model is (33+22)/412 = **13.4%**

The **sensitivity** for the **LDA** model is (154/176) = **87.6%**

The **testing error** for the **QDA** model is (28+18)/412 = **11.2%**

The **sensitivity** for the **QDA** model is (158/176) = **89.8%**

and a reminder that the **testing error and sensitivity** for our **logistic regression** model (θ = 0.5) from question 2 is **13.6%** & **86.4%**, respectively.

By comparing the testing error and sensitivity for these three models, I would recommend the **QDA** model for this problem as it not only has the lowest testing error of the three (11.2%), it also has the highest sensitivity (89.8%). This means that, of the three models, the QDA is the most effective at identifying forged banknotes. This decision and recommendation are made under the assumption that there is a high-cost false negative vs low cost false positive dynamic present in this scenario, and the bank/other financial institution in question is most concerned with identifying forged banknotes. This context, or domain knowledge (even if it is, in this case, assumed) helps inform us on the most appropriate model to use in a particular situation.

4. **Consider a binary classification problem Y ε {0,1} with one predictor X. Assume that X is normally distributed ($X \sim N(\mu,\sigma^2)$) in each class with $X \sim N(0,4)$ in class 0 and $X \sim N(2,4)$ in class 1. Calculate Bayes error rate when the prior probability of being in class 0 is $\pi_0 = 0.4$.**

To calculate Bayes error rate, we first need to find Bayes decision boundary.

Daniel Rance
49144736
dra124

The normal density function takes the form:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}}\ exp\left(-\frac{1}{2\sigma^2}(x - \mu)^2\right)$$

And with the class information we have been provided:

$$0: X \sim N(0,4) \rightarrow f_0(x) = \frac{1}{\sqrt{4\pi}}\ exp\left(-\frac{1}{8}(x)^2\right) :\ \pi_0 = 0.4$$

$$1: X \sim N(2,4) \rightarrow f_1(x) = \frac{1}{\sqrt{4\pi}}\ exp\left(-\frac{1}{8}(x-2)^2\right) :\ \pi_1 = 0.6$$

The decision boundary is given by:

$$\pi_0 f_0(x) = \pi_1 f_1(x)$$

Which can be rearranged as:

$$\pi_0 \frac{1}{\sqrt{4\pi}}\ exp\left(-\frac{1}{8}(x)^2\right) = \pi_1 \frac{1}{\sqrt{4\pi}}\ exp\left(-\frac{1}{8}(x-2)^2\right)$$

Cancelling out the common terms, we receive:

$$\pi_0\ exp\left(-\frac{1}{8}(x)^2\right) = \pi_1\ exp\left(-\frac{1}{8}(x-2)^2\right)$$

To deal with the exponents, we can apply natural logs:

$$ln\left(\pi_0\ exp\left(-\frac{1}{8}(x)^2\right)\right) = ln\left(\pi_1\ exp\left(-\frac{1}{8}(x-2)^2\right)\right)$$

Which is equivalent to:

$$\ln(\pi_0) + ln\left(exp\left(-\frac{1}{8}(x)^2\right)\right) = \ln(\pi_1) + \ln\left(exp\left(-\frac{1}{8}(x-2)^2\right)\right)$$

We can now remove the exponents:

$$\ln(\pi_0) - \frac{1}{8}x^2 = \ln(\pi_1) - \frac{1}{8}(x-2)^2$$

And subtract $\ln(\pi_1)$ from both sides:

Daniel Rance
49144736
dra124

$$\ln(\pi_0) - \ln(\pi_1) - \frac{1}{8}x^2 = -\frac{1}{8}(x-2)^2$$

Applying another property of logs:

$$\ln\left(\frac{\pi_0}{\pi_1}\right) - \frac{1}{8}x^2 = -\frac{1}{8}(x-2)^2$$

We can expand the right side of the equation to show:

$$= -\frac{1}{8}(x^2 - 4x + 4)$$

Which then becomes:

$$\ln\left(\frac{\pi_0}{\pi_1}\right) - \frac{1}{8}x^2 = -\frac{1}{8}x^2 + \frac{1}{2}x - \frac{1}{2}$$

After some cancelation:

$$\ln\left(\frac{\pi_0}{\pi_1}\right) = \frac{1}{2}x - \frac{1}{2}$$

And, finally, to solve for x:

$$\frac{1}{2}x = \ln\left(\frac{\pi_0}{\pi_1}\right) + \frac{1}{2}$$

$$x = 2 \cdot \ln\left(\frac{\pi_0}{\pi_1}\right) + 1 = bound$$

as the decision boundary. Bayes error rate is given by:

$$\pi_0 P(X > bound | Y = 0) + \pi_1 P(X < bound | Y = 1)$$

Which we can calculate using the following R code:

```
X = seq(-4, 8, length = 100)
#class 0
pi_0 = 0.4
f_0 = dnorm(X, mean = 0, sd = 2)
#class 1
pi_1 = 0.6
f_1 = dnorm(X, mean = 2, sd = 2)
#bayes decision boundary
bayes_boundary = 2*log(pi_0 / pi_1) + 1
bayes_error_rate = pi_0 * (1 - pnorm(bayes_boundary, mean = 0,
sd = 2)) + pi_1 * pnorm(bayes_boundary, mean = 2, sd = 2)

bayes_error_rate
## [1] 0.2945026
```

Therefore, our error rate for this problem is **0.294**.