Assignment 4

Mike Whitley

16/05/2021

Extra code has been put at bottom after question 6 for ease of reading where possible, whilst still keeping meaning

A population of women who were at least 21 years old, of Pima Indian heritage and living near Phoenix, Arizona, was tested for diabetes according to World Health Organization criteria. The data were collected by the US National Institute of Diabetes and Digestive and Kidney Diseases.

Question one

Fit a Linear Discriminant Analysis model (function lda in package MASS) and calculate the misclassification error on the provided test set. (2 marks)

> The misclassification error for LDA based on the testing/training data is 0.19 or 19%

Question two

Which kind of misclassification is more common in the test data: patients with diabetes misclassified as healthy, or healthy patients misclassified as having diabetes? (2 marks)

> The category type equals 1 if diabetes is present and 0 otherwise.
>
> Based on the table we can see that 4 people are false positive where they are predicted to have diabetes but don't have it whilst 15 people are false negative meaning they were predicted to not have diabetes however they do have it. Therefore, more patients were misclassified as not having diabetes when they infact do have diabetes

Question three

Fit a Quadratic discriminant Analysis model (function qda in package MASS). Write down the misclassification error (Question 1) for the QDA model. (1 marks)

> The misclassification error of QDA model based on the testing/training data is 0.27 or 27%

Question four

A health organization wants you to recommend one of the two models for diagnosing diabetes. What would you tell them? Explain your decision in a way that a non-statistician could understand. (2 marks)

> Based on the amount of data we have here and the results I would recommend the Linear Discriminant Analysis model. We can see that when using the LDA model we have an accuracy rate of 81% meaning we correctly identify if the individual has diabetes 81 out of 100 times, compared to the QDA model being only 73 out of 100 times. However as this is a health-related issue of high important I would recommend gathering further data as Linear discriminant analysis preforms better with low observations and you may be able to further increase the predication percentage with more data to analyze.

Question five

Fit a logistic regression model. What is the test error for this model? (1 mark)

> The test error for logistic regression is 19% or 0.19

Question six

A woman wants to know about her diabetes status. Following data is available: npreg glu bp skin bmi ped age 5 111 81 33 25.1 0.36 48 Predict the diabetes status by logistic regression without using the R function predict. (And outline the calculations involved) (2 marks)

Code has been left in as asks for calculations involved

```
(glm.fit$coefficients) ##gives me my coefficents

##   (Intercept)         npreg           glu            bp          skin 
## -9.2277085819  0.1071682885  0.0196674170 -0.0006221187  0.0081414644 
##           bmi           ped           age 
##  0.1067457276  1.6743607820  0.0331500973

int = -9.2277085819
npregCoef = 0.1071682885
gluCoef = 0.0196674170
bpCoef = -0.0006221187
skinCoef = 0.0081414644
bmiCoef = 0.1067457276
pedCoef = 1.6743607820
ageCoef = 0.0331500973
```

```r
#new values introduced
npreg = 5
glu = 111
bp = 81
skin = 33
bmi = 25.1
ped = 0.36
age = 48

#we just input the variables into the equation
Answer = int +
  npregCoef * npreg +
  gluCoef * glu +
  bpCoef * bp +
  skinCoef * skin +
  bmiCoef * bmi +
  pedCoef * ped +
  ageCoef * age

paste0("My B0 + XB is  ", (Answer))
```

```
## [1] "My B0 + XB is  -1.41721482722"
```

```r
#the equation to work out prediction
My_prediction <- function(glm_train_output) {
  return (exp(glm_train_output)) / (1 + exp(glm_train_output))
}
paste0("My resulting prediction is ", My_prediction(Answer))
```

```
## [1] "My resulting prediction is 0.24238817057889"
```

For output predictions lower than 0.5 she will be classified as non-Diabetic and greater than 0.5 she will be classified as diabetic in this case the prediction is 0.24 as this is less than 0.5 we predict that she is non-Diabetic.

Appendices

1)

```
#install.packages("DAAG")
#install.packages("caret")

pima_test <- read.table("pima_test.txt", sep = "" , header = T , nrows = 100,
                     na.strings ="", stringsAsFactors= F)
pima_train <- read.table("pima_train.txt", sep = "" , header = T , nrows = 10
0,
                     na.strings ="", stringsAsFactors= F)
head(pima_test)

##   npreg glu bp skin  bmi   ped age type
## 1     5  86 68   28 30.2 0.364  24    0
## 2     7 195 70   33 25.1 0.163  55    1
## 3     5  77 82   41 35.8 0.156  35    0
## 4     0 165 76   43 47.9 0.259  26    0
## 5     0 107 60   25 26.4 0.133  23    0
## 6     5  97 76   27 35.6 0.378  52    1

head(pima_train)

##   npreg glu bp skin  bmi   ped age type
## 1     6 148 72   35 33.6 0.627  50    1
## 2     1  85 66   29 26.6 0.351  31    0
## 3     1  89 66   23 28.1 0.167  21    0
## 4     3  78 50   32 31.0 0.248  26    1
## 5     2 197 70   45 30.5 0.158  53    1
## 6     5 166 72   19 25.8 0.587  51    1

2)
```

for misclassification we want to use bayes classifier

```
library(MASS)
library(DAAG) #required for confusion matrix

## Loading required package: lattice

##
## Attaching package: 'DAAG'

## The following object is masked from 'package:MASS':
##
##     hills

table(pima_train$type) #use this to work out 64 36 split for lda 1/k
```

```
## 
##  0  1 
## 64 36
```

```
attach(pima_train)
lda_model <- lda(type ~ npreg + glu + bp + skin + bmi + ped + age, data=pima_
train,         #completes LDA analysis
         prior=c(0.64, 0.36))  #0.36 and 0.64 is class proportion
detach(pima_train)
lda_model
```

```
## Call:
## lda(type ~ npreg + glu + bp + skin + bmi + ped + age, data = pima_train, 
##     prior = c(0.64, 0.36))
## 
## Prior probabilities of groups:
##    0    1 
## 0.64 0.36 
## 
## Group means:
##      npreg      glu       bp     skin      bmi       ped      age
## 0 3.093750 107.5469 69.09375 27.34375 31.05625 0.4318906 29.17188
## 1 4.916667 136.4722 74.55556 35.30556 37.55833 0.6543611 37.05556
## 
## Coefficients of linear discriminants:
##                LD1
## npreg  0.061095011
## glu    0.014057484
## bp    -0.002200395
## skin   0.006869057
## bmi    0.072974448
## ped    0.997322339
## age    0.029958760
```

```
attach(pima_test)

#testing_predict <- predict(lda_model, data=pima_test) #Classify multivariate
observations in conjunction with lda, and also project data onto the linear d
iscriminants.


#table(Predicted=testing_predict$class, Type=type)      #this table should gi
ve me the data needed to calculate accuracy i.e error rate
# The error rate is simply the number of misclassifications divided by the to
tal sample size.so can check with using the table
#\\
#table(pima_train$type, testing_predict$class)

#library(caret)
#confusionMatrix(testing_predict$class, pima_test$type)
```

```
pred_value <- predict(lda_model, newdata=pima_test[,])$class
table(Type=pima_test$type, pred_value)

##     pred_value
## Type  0  1
##    0 63  4
##    1 15 18

confusion(pima_test$type, pred_value) #81% accuracy so 19% misclassification

## Overall accuracy = 0.81
##
## Confusion matrix
##       Predicted (cv)
## Actual  [,1]  [,2]
##   [1,] 0.940 0.060
##   [2,] 0.455 0.545

#can check with
#15 + 4 / 100 = 0.19 or 19%
detach(pima_test)
```

3)

```
attach(pima_train)
qda_model <- qda(type ~ npreg + glu + bp + skin + bmi + ped + age, data=pima_
train,         #completes LDA analysis
         prior=c(0.64, 0.36))

pred_value_qda <- predict(qda_model, newdata=pima_test[,])$class
table(Type=pima_test$type, pred_value_qda)

##     pred_value_qda
## Type  0  1
##    0 55 12
##    1 15 18

confusion(pima_test$type, pred_value_qda) #0.27%

## Overall accuracy = 0.73
##
## Confusion matrix
##       Predicted (cv)
## Actual  [,1]  [,2]
##   [1,] 0.821 0.179
##   [2,] 0.455 0.545

#confirm with 15+12 / 100 = 0.27
detach(pima_train)
```

4) No Code

5)

```
glm.fit=glm(type~npreg + glu + bp + skin + bmi + ped + age, data=pima_train,
family=binomial)
summary(glm.fit)

## 
## Call:
## glm(formula = type ~ npreg + glu + bp + skin + bmi + ped + age, 
##     family = binomial, data = pima_train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.9015  -0.7049  -0.3487   0.6975   2.2940  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -9.2277086  2.2035735  -4.188 2.82e-05 ***
## npreg        0.1071683  0.0996790   1.075   0.2823    
## glu          0.0196674  0.0093354   2.107   0.0351 *  
## bp          -0.0006221  0.0202762  -0.031   0.9755    
## skin         0.0081415  0.0337234   0.241   0.8092    
## bmi          0.1067457  0.0476649   2.240   0.0251 *  
## ped          1.6743608  0.8740195   1.916   0.0554 .  
## age          0.0331501  0.0327437   1.012   0.3113    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 130.68  on 99  degrees of freedom
## Residual deviance:  90.77  on 92  degrees of freedom
## AIC: 106.77
## 
## Number of Fisher Scoring iterations: 5

glm.probs=predict(glm.fit, newdata = pima_test, type="response")
glm.probs[1:10] #check the data for 10 entries

##          1          2          3          4          5          6 
7 
## 0.10101279 0.58861182 0.16087406 0.67456373 0.04093961 0.38823727 0.118700
97 
##          8          9         10 
## 0.36157186 0.46028678 0.81090892
```

```
glm.pred=rep("No Diabeties",100)
glm.pred[glm.probs>.5]="Has Diabeties"
table(glm.pred, pima_test$type)
```

```
##
## glm.pred         0  1
##   Has Diabeties  4 18
##   No Diabeties  63 15
```

```
#checking the testing error
confusion(pima_test$type, glm.pred) #0.19%
```

```
## Overall accuracy = 0.19
##
## Confusion matrix
##       Predicted (cv)
## Actual  [,1]  [,2]
##   [1,] 0.060 0.940
##   [2,] 0.545 0.455
```

```
#4+15 / 100 = 0.19 = 19%
```