MATH9952 Modern Applied Statistical Models
ASSIGNMENT MARKING SCHEME

Student No. D16126734 Name: JERRY KIELY

Assignment 2: Logistic Regression.

59%

Part 1: R Code	~~20~~ 24
Comments: very good code used to identify a model. You might have used customised tests to explore the CI's and OR's of the final model though.	

Part 2: Report	15
Comments: A very terse report that needs more explanatory text - you need to more fully discuss what you did, why you did it and your results. You need to fully discuss/interpret the final model.	

Part 3: Predicted Probability	20
Part 4: Algebra	0
TOTAL	59 %

The Retention Dataset

Jerry Kiely

27 March 2017

Introduction

This analysi is concerned with data relating to student retention in the Engineering faculty of DIT. The purpose of the analysis will be to use logistic regression models to identify and quantify relevant risk factors in student retention. The data includes risk factors regarding prior academic performance (e.g. leaving certificate results, leaving certificate maths grade), and personal characteristics (gender, home address, CAO choices made, etc.)

Variable name	Details
passed	Whether the student qualified to enter second year of their degree (0 = did not qualify, 1 = qualified)
gender	Male (1) or Female (0)
lc_points	Leaving certificate points achieved
mathgrd	Leaving certificate mathematics grade
CAO_choice	CAO ranked choice of programme entered
Address	Coded home address; 1 = Dublin, 2 = Dublin commuter belt, 3 = outside Dublin commuter belt

The Data

The data contains some extra columns that we don't need.

```
colnames(retention)
```

```
## [1] "X"          "gender"     "passed"     "mathgrd"    "CAO_choice"
## [6] "address"    "lc_points"  "lc_points.1"
```

so we remove them:

```
retention$X           = NULL
retention$lc_points.1 = NULL
```

we convert columns to factors:

```
retention$mathgrd     = as.factor(retention$mathgrd)
retention$address     = as.factor(retention$address)
```

and we remove rows where NULLs or NAs are present:

```
retention             = retention[complete.cases(retention),]
```

finally we have a look at the data:

```
head(retention)
```

```
##   gender passed mathgrd CAO_choice address lc_points
## 1      0      0     80+          1       2       315
## 2      0      0     >20          1       1       270
## 3      0      1   50-60          1       2       370
```

```
## 4     0     1   20-30          2        2        295
## 6     0     0   20-30          1        1        260
## 7     0     1   35-45          1        1        280
```

The Model

First thing we do is fit a linear model to the data, including all possible interactions between the predictors.

Using either of the drop1 or the step functions we prune unimportant predictors from the model.

```
## Single term deletions
##
## Model:
## passed ~ gender + mathgrd + CAO_choice + lc_points + gender:CAO_choice
##                   Df Deviance    AIC     LRT Pr(>Chi)
## <none>                 288.38 316.38
## mathgrd            5   303.06 321.07 14.6893 0.011776 *
## lc_points          1   296.24 322.24  7.8668 0.005035 **
## gender:CAO_choice  3   296.23 318.23  7.8501 0.049212 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I don't understand why this is included and other interactions are not? is this what is left after "pruning"?

We are left with the following formula:

```
##
## Call:
## glm(formula = passed ~ gender + mathgrd + CAO_choice + lc_points +
##     gender:CAO_choice, family = binomial(link = "logit"), data = retention)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9148  -1.1097   0.6604   0.9128   2.0125
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -3.358323   1.038658  -3.233 0.001224 **
## gender              0.082133   0.482321   0.170 0.864784
## mathgrd20-30        1.224932   0.553829   2.212 0.026984 *
## mathgrd35-45        1.958951   0.582856   3.361 0.000777 ***
## mathgrd50-60        1.210271   0.592484   2.043 0.041081 *
## mathgrd65-75        0.753588   0.728172   1.035 0.300714
## mathgrd80+          1.350320   0.919415   1.469 0.141921
## CAO_choice2         2.749213   1.239096   2.219 0.026505 *
## CAO_choice3         1.187253   1.331843   0.891 0.372695
## CAO_choice4         1.114424   1.602384   0.695 0.486755
## lc_points           0.007958   0.002914   2.731 0.006322 **
## gender:CAO_choice2 -3.087517   1.291805  -2.390 0.016845 *
## gender:CAO_choice3 -0.832331   1.411712  -0.590 0.555466
## gender:CAO_choice4 -1.438172   1.649171  -0.872 0.383177
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 331.05  on 247  degrees of freedom
```

```
## Residual deviance: 288.38  on 234  degrees of freedom
## AIC: 316.38
##
## Number of Fisher Scoring iterations: 4
```

looking at the coefficient for lc_points, also it's log odds ratio for example, we see a value of 0.0079582 with an odds ratio of 1.00799 which would indicate that the odds of the student entering the second year of their degree would increase by 1.00799 for every leaving certificate points achieved.

Explanation of the

discuss the other predictors included in the model?

```
setwd("~/Workspace/College/DIT/MATH9952/Data")

# read in the data
retention = read.csv("retention.csv", header = T)

# remove unnecessary columns
retention$X            = NULL
retention$lc_points.1 = NULL

# convert columns to factore
retention$mathgrd      = as.factor(retention$mathgrd)
retention$address      = as.factor(retention$address)

# remove rows where NULLs or NAs are present
retention              = retention[complete.cases(retention),]

# have a look at the data
head(retention)

attach(retention)

# list the column names
colnames(retention)



# Question 1

# fit a model with all interractions
fit1 = glm(passed ~ .*., family = binomial(link = "logit"), data = retention)
summary(fit1)

# prune unnecessary predictors
step(fit1, scope = list(lower = ~ 1, upper = ~ .*.), direction = "backward", trace = 1)

# fit the final model
fitf = glm(passed ~ gender + mathgrd + CAO_choice + lc_points + gender:CAO_choice, family = binomial(link =
"logit"), data = retention)
summary(fitf)



# Question 3

nd = data.frame(gender = 1, lc_points = 300, mathgrd = "50-60", CAO_choice = "3")
p  = predict(fitf, newdata = nd, se = T)

prob = exp(p$fit) / (1 + exp(p$fit))
ciu  = exp(p$fit + 1.96 * p$se.fit) / (1 + exp(p$fit + 1.96 * p$se.fit))
cil  = exp(p$fit - 1.96 * p$se.fit) / (1 + exp(p$fit - 1.96 * p$se.fit))

data.frame(prob = prob, upperCI = ciu, lowerCI = cil)
#        prob   upperCI   lowerCI
# 1 0.6629496 0.8398823 0.4244774


# Question 4

fitz = glm(passed ~ gender + lc_points, family = binomial(link = "logit"), data = retention)
summary(fitz)

# NR Method

x1      = gender
x2      = lc_points
y       = passed

beta0  = 1
beta1  = 0
beta2  = 0
beta   = matrix(c(beta0, beta1, beta2), nrow = 3)
```

```
# iterations start...

eta      = (beta[1, 1] + beta[2, 1] * x1 + beta[3, 1] * x2)

score1  = sum(y        - ((1 * exp(eta)) / (1 + exp(eta))))
score2  = sum((y * x1) - ((1 * x1 * exp(eta)) / (1 + exp(eta))))
score3  = sum((y * x2) - ((1 * x2 * exp(eta)) / (1 + exp(eta))))

h11      = sum( (1 * exp(eta)^2)            / (1 + exp(eta))^2 - ((1 * exp(eta)) / (1 + exp(eta))))
h12      = sum( (1 * x1 * exp(eta)^2)       / (1 + exp(eta))^2 - ((1 * x1 * exp(eta)) / (1 + exp(eta))))
h13      = sum( (1 * x2 * exp(eta)^2)       / (1 + exp(eta))^2 - ((1 * x2 * exp(eta)) / (1 + exp(eta))))
h22      = sum( (1 * x1^2 * exp(eta)^2)     / (1 + exp(eta))^2 - ((1 * x1^2 * exp(eta)) / (1 + exp(eta))))
h23      = sum( (1 * x1 * x2 * exp(eta)^2) / (1 + exp(eta))^2 - ((1 * x1 * x2 * exp(eta)) / (1 + exp(eta))))
h33      = sum( (1 * x2^2 * exp(eta)^2)     / (1 + exp(eta))^2 - ((1 * x2^2 * exp(eta)) / (1 + exp(eta))))

u        = matrix(c(score1, score2, score3), nrow = 3)
h        = matrix(c(h11, h12, h13, h12, h22, h23, h13, h23, h33), nrow = 3, byrow = T)

betanew = beta - solve(h) %*% u
beta     = betanew

result  = data.frame(beta = beta, score = u, hessian = h)
result

# iterations end...
```

```
# Appendix

# fit a model without interractions - simpler!

# fit1  = glm(passed ~ ., family = binomial(link = "logit"), data = retention)
# formula(fit1)
# summary(fit1)

# use the step function to help you find the predictors to drop...

# step(fit1, scope = list(lower = ~ 1, upper = ~ .), direction = "backward", trace = 1)

# or use the drop1 function to manually drop predictors...

# drop1(fit1, test = 'LRT')
# fit2 = update(fit1, ~. - CAO_choice)
# summary(fit2)
#
# drop1(fit2, test = 'LRT')
# fit3 = update(fit2, ~. - address)
# summary(fit3)
#
# drop1(fit3, test = 'LRT')
```

Jerry Kiely

Q4

20/20

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_{gender} \\ \beta_{lcpoints} \end{pmatrix} \qquad x = \begin{pmatrix} 1 \\ x_{gender} \\ x_{lcpoints} \end{pmatrix}$$

$$\eta_i = x_i' \beta$$

$$L(\beta) = \prod_{i=1}^{n} \binom{n_i}{y_i} \frac{(e^{\eta_i})^{y_i}}{(1+e^{\eta_i})^{n_i}}$$

$$\ell(\beta) = \sum_{i=1}^{n} \left(\ln \binom{n_i}{y_i} + y_i \eta_i - n_i \ln (1+e^{\eta_i}) \right)$$

$$\left[\binom{1}{1} = 1 \Rightarrow \binom{n_i}{y_i} = 1 \right]$$

$$\left[\begin{array}{l} n_i = 1 \\ y_i = 1 \end{array} \right]$$

$$U = \begin{pmatrix} \dfrac{\partial \ell(\beta)}{\partial \beta_0} \\ \dfrac{\partial \ell(\beta)}{\partial \beta_1} \\ \dfrac{\partial \ell(\beta)}{\partial \beta_2} \end{pmatrix}$$

$$\frac{\partial \ell(\beta)}{\partial \beta_0} = \sum_{i=1}^{n} \left(y_i - \frac{e^{\eta_i}}{1+e^{\eta_i}} \right)$$

$$\frac{\partial \ell(\beta)}{\partial \beta_{gender}} = \sum_{i=1}^{n} \left(y_i \cdot x_{gender} - \frac{e^{\eta_i} \cdot x_{gender}}{1+e^{\eta_i}} \right)$$

$$\frac{\partial l(\beta)}{\partial \beta_{lcpoints}} = \sum_{i=1}^{n} \left(y_i \cdot x_{lcpoints} - \frac{e^{\eta_i} \cdot x_{lcpoints}}{1+e^{\eta_i}} \right)$$

$$H = \begin{pmatrix} \frac{\partial^2 l(\beta)}{\partial \beta_0^2} & \frac{\partial^2 l(\beta)}{\partial \beta_0 \beta_g} & \frac{\partial^2 l(\beta)}{\partial \beta_0 \beta_e} \\ \frac{\partial^2 l(\beta)}{\partial \beta_g \beta_0} & \frac{\partial^2 l(\beta)}{\partial \beta_g^2} & \frac{\partial^2 l(\beta)}{\partial \beta_g \beta_e} \\ \frac{\partial^2 l(\beta)}{\partial \beta_e \beta_0} & \frac{\partial^2 l(\beta)}{\partial \beta_e \beta_g} & \frac{\partial^2 l(\beta)}{\partial \beta_e^2} \end{pmatrix}$$

$$\frac{\partial^2 l(\beta)}{\partial \beta_0^2} = \frac{(e^{\eta_i})^2}{(1+e^{\eta_i})^2} - \frac{e^{\eta_i}}{(1+e^{\eta_i})}$$

$$\frac{\partial^2 l(\beta)}{\partial \beta_0 \beta_g} = \frac{(e^{\eta_i})^2 \cdot x_{gender}}{(1+e^{\eta_i})^2} - \frac{e^{\eta_i} \cdot x_{gender}}{(1+e^{\eta_i})}$$

$$\frac{\partial^2 l(\beta)}{\partial \beta_0 \beta_e} = \frac{(e^{\eta_i})^2 x_{lcpoints}}{(1+e^{\eta_i})^2} - \frac{e^{\eta_i} x_{lcpoints}}{(1+e^{\eta_i})}$$

$$\frac{\partial^2 l(\beta)}{\partial \beta_g^2} = \frac{(e^{\eta_i})^2 (x_{gender})^2}{(1+e^{\eta_i})^2} - \frac{e^{\eta_i} (x_{gender})^2}{(1+e^{\eta_i})}$$

$$\frac{\partial^2 l(\beta)}{\partial \beta_g \beta_e} = \frac{(e^{\eta_i})^2 \cdot x_{gender} \cdot x_{lcpoints}}{(1+e^{\eta_i})^2} - \frac{e^{\eta_i} \cdot x_{gender} \cdot x_{lcpoints}}{(1+e^{\eta_i})}$$

$$\frac{\partial\, l(\beta)}{\partial \beta_x^2} = \frac{(e^{\eta_i})^2 . (x_{e(points)})^2}{(1+e^{\eta_i})^2} - \frac{e^{\eta_i} . (x_{e(points)})^2}{(1+e^{\eta_i})}$$

Classic Newton - Raphson, of the form

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

can be used to find the roots of the equation

$$f(x) = 0$$

we may apply the multivariable approach to our log likelihood expressions in order to find the maximum values of the β's

$$\beta_{i+1} = \beta_i - H_i^{-1} u_i$$

starting with estimates for β, we iterate the above until we reach convergence - i.e consistency to within 5 decimal places between iterations for example.