Lab4_Answer

Anyi Guo

23/10/2018

Lab 4 Logistic Regression

Problem Statement

A researcher is interested in how variables, such as GRE (Graduate Record Exam scores), GPA (grade point average) and rank (prestige of the undergraduate institution), affect admission into graduate school. The response variable, admit/do not admit, is a binary variable.

Dataset

The dataset is included in the package aod. Install the package and include package using the command library(aod).

```
library(aod)
mydata<-read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
# take a look at the first few rows
head(mydata)
```

```
##   admit gre  gpa rank
## 1     0 380 3.61    3
## 2     1 660 3.67    3
## 3     1 800 4.00    1
## 4     1 640 3.19    4
## 5     0 520 2.93    4
## 6     1 760 3.00    2
```

Using the following command to load the dataset ## admit gre gpa rank ##1 0 380 3.61 3 ##2 1 660 3.67 3 ##3 1 800 4.00 1 ##4 1 640 3.19 4 ##5 0 520 2.93 4 ##6 1 760 3.00 2

[More on reading and writing CSV files, see here: https://swcarpentry.github.io/r-novice-inflammation/11-supp-read-write-csv/index.html]

This dataset has a binary response (outcome, dependent) variable called admit. There are three predictor variables: gre, gpa and rank. We will treat the variables gre and gpa as continuous. The variable rank takes on the values 1 through 4. Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest.

Questions

1) Get basic descriptives for the entire data set using summary(). View the dataset using View().

```
summary(mydata)
```

```
##      admit             gre             gpa             rank
##  Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000
##  Median :0.0000   Median :580.0   Median :3.395   Median :2.000
##  Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
##  3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
##  Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000
```

```
View(mydata)
```

2) How many observations are there in this dataset?

```
dim(mydata)
```

```
## [1] 400   4
```

400 observations of 4 rows.

3) Get the standard deviations for the first three variables (i.e., admit, gre and gpa). Hint: use sapply to apply the sd function to each variable in the dataset: sapply(mydata, sd). Now get the mean admit, gre and gpa in a similar way.

```
sapply(mydata[,-4],sd)
```

```
##       admit         gre         gpa
##   0.4660867 115.5165364   0.3805668
```

```
sapply(mydata[,-4],mean)
```

```
##    admit      gre      gpa
##   0.3175 587.7000   3.3899
```

Using `[,-4]` to ignore the fourth column which is `rank`. Ignoring it as it is a categorical column, so these numbers are not meaningful to them.

4) Convert rank to a factor to indicate that rank should be treated as a categorical variable. (Hint: use factor() function) [More on factors, see the tutorial here: https://swcarpentry.github.io/r-novice-inflammation/12-supp-factors/ index.html]

```
mydata$rank<-factor(mydata$rank)
```

5) Estimate a logistic regression model using the glm function, and get the results using the summary command.

```
glm.fit<-glm(admit~gre+gpa+rank,data=mydata,family = binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa + rank, family = binomial, data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6268  -0.8662  -0.6388   1.1490   2.0790
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.989979   1.139951  -3.500 0.000465 ***
## gre          0.002264   0.001094   2.070 0.038465 *
## gpa          0.804038   0.331819   2.423 0.015388 *
## rank2       -0.675443   0.316490  -2.134 0.032829 *
## rank3       -1.340204   0.345306  -3.881 0.000104 ***
## rank4       -1.551464   0.417832  -3.713 0.000205 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 458.52  on 394  degrees of freedom
## AIC: 470.52
##
## Number of Fisher Scoring iterations: 4
```

6) Do you notice variable rank is replaced with categorical variables rank2, rank3, and rank4 that can only take values of 0 or 1? Recall that the original variable rank can take values of 1, 2, 3, or 4. Why isn't a variable rank1 needed? If rank is 1, what are the values of rank2, rank3 and rank4?

If `rank` is 1, then `rank2` is 0, `rank3` is 0 and `rank4` is 0.

If `rank` is 2, then `rank2` is 1, `rank3` and `rank4` are 0.

If `rank` is 3, then `rank2` is 0, `rank3` is 1 and `rank4` is 0.

If `rank` is 4, then `rank2` and `rank3` are 0 and `rank4` is 1.

7) From the z-statistics and p-values of the variables, report which variables are statistically significant.

The z-statistics of all the variables are large and the p-values of all the variables are small (<0.05). All the variables are statistically significant.

8) Use the model to predict the training dataset and store the results to a vector of probabilities admit.prob.

```
admit.probs <- predict(glm.fit,type="response")
admit.probs
```

```
##          1          2          3          4          5          6
## 0.17262654 0.29217496 0.73840825 0.17838461 0.11835391 0.36996994
##          7          8          9         10         11         12
## 0.41924616 0.21700328 0.20073518 0.51786820 0.37431440 0.40020025
##         13         14         15         16         17         18
## 0.72053858 0.35345462 0.69237989 0.18582508 0.33993917 0.07895335
##         19         20         21         22         23         24
## 0.54022772 0.57351182 0.16122101 0.43727108 0.12837525 0.19204860
##         25         26         27         28         29         30
## 0.43759396 0.68229503 0.57848091 0.20475422 0.42307349 0.45829857
##         31         32         33         34         35         36
## 0.21765393 0.28583616 0.22481919 0.42494837 0.34296523 0.21293277
##         37         38         39         40         41         42
## 0.48413281 0.13931720 0.26569575 0.11942769 0.18975965 0.33567002
##         43         44         45         46         47         48
## 0.31560404 0.17702923 0.32817441 0.18025548 0.36121718 0.11699101
##         49         50         51         52         53         54
## 0.07235381 0.15047417 0.31488795 0.11624726 0.23936553 0.37838478
##         55         56         57         58         59         60
## 0.24045684 0.39213236 0.18283980 0.10853139 0.30472142 0.12837525
##         61         62         63         64         65         66
## 0.33078459 0.16742893 0.28289780 0.33295972 0.30988311 0.39645173
##         67         68         69         70         71         72
## 0.27784995 0.51681586 0.57206626 0.69436828 0.33966212 0.07486000
##         73         74         75         76         77         78
## 0.15073716 0.46607599 0.24284830 0.38139149 0.20415281 0.42494837
##         79         80         81         82         83         84
## 0.43570986 0.65251556 0.16456653 0.31150713 0.20517359 0.08776685
##         85         86         87         88         89         90
## 0.21358749 0.25126279 0.34584314 0.37549461 0.55783057 0.51131037
##         91         92         93         94         95         96
## 0.49978497 0.63809471 0.57000341 0.26968427 0.40010880 0.37907977
##         97         98         99        100        101        102
## 0.22063013 0.33002244 0.31762762 0.14640896 0.11633954 0.24114689
##        103        104        105        106        107        108
## 0.11883427 0.28100436 0.50126183 0.35394219 0.61241920 0.25695415
##        109        110        111        112        113        114
## 0.11218813 0.30904921 0.17869743 0.13603549 0.10881750 0.48942091
##        115        116        117        118        119        120
## 0.35153649 0.32780508 0.29004920 0.47768876 0.68922540 0.09863460
##        121        122        123        124        125        126
## 0.38205848 0.19283124 0.13456621 0.14161529 0.35890251 0.16784107
##        127        128        129        130        131        132
## 0.55353632 0.29761787 0.29364378 0.12270194 0.32900715 0.27429792
##        133        134        135        136        137        138
## 0.35016196 0.15167362 0.26397051 0.20956391 0.16855273 0.37076538
##        139        140        141        142        143        144
## 0.37104174 0.56147017 0.48592324 0.24487554 0.27496207 0.21702497
##        145        146        147        148        149        150
## 0.18326999 0.15292361 0.30053113 0.13202601 0.36278299 0.58590453
##        151        152        153        154        155        156
## 0.69607194 0.26076336 0.48793196 0.22533437 0.27701027 0.12691355
##        157        158        159        160        161        162
## 0.20243105 0.49385024 0.40979572 0.33767745 0.31214097 0.40081797
##        163        164        165        166        167        168
## 0.44572710 0.21536268 0.33209361 0.69237989 0.12564635 0.33881603
##        169        170        171        172        173        174
## 0.27253083 0.25713529 0.16766865 0.13610230 0.27045353 0.47601029
##        175        176        177        178        179        180
## 0.17207711 0.36543032 0.20079352 0.20929210 0.22290898 0.09702710
##        181        182        183        184        185        186
## 0.29173405 0.21592659 0.53390445 0.41213948 0.10284874 0.51016205
##        187        188        189        190        191        192
## 0.23875288 0.26184001 0.28313813 0.30160149 0.29894660 0.33797096
##        193        194        195        196        197        198
## 0.29780561 0.14252603 0.37361105 0.37499458 0.20306181 0.11520619
##        199        200        201        202        203        204
## 0.25867413 0.23203530 0.29790835 0.31450637 0.69237989 0.19176895
##        205        206        207        208        209        210
## 0.62160882 0.37552455 0.62994688 0.59336886 0.17269671 0.36867073
##        211        212        213        214        215        216
## 0.23500145 0.28417171 0.21145148 0.23806753 0.39069474 0.18303592
##        217        218        219        220        221        222
## 0.29144726 0.49458858 0.36532833 0.37499458 0.18691983 0.35841190
##        223        224        225        226        227        228
## 0.38346629 0.32549498 0.37234438 0.29200523 0.40539785 0.13119209
##        229        230        231        232        233        234
## 0.30562595 0.42917277 0.17040039 0.20845157 0.25212831 0.09688336
##        235        236        237        238        239        240
## 0.65921863 0.30806878 0.40979572 0.41039144 0.10815929 0.27465027
##        241        242        243        244        245        246
## 0.19001218 0.56239934 0.19616746 0.33794240 0.41996550 0.40736827
##        247        248        249        250        251        252
## 0.39171070 0.24596016 0.29657173 0.29278619 0.20011793 0.17414395
##        253        254        255        256        257        258
## 0.43247252 0.18780755 0.26200847 0.23371984 0.30267400 0.32075797
##        259        260        261        262        263        264
## 0.33944941 0.46187255 0.34863249 0.24298996 0.16969339 0.32075797
##        265        266        267        268        269        270
## 0.26562483 0.14378335 0.15865328 0.26021896 0.41492493 0.12579904
##        271        272        273        274        275        276
## 0.48994106 0.19310678 0.45641226 0.54337733 0.27302605 0.28684953
##        277        278        279        280        281        282
## 0.22143462 0.55028996 0.16945136 0.34384116 0.49925174 0.13172559
##        283        284        285        286        287        288
## 0.21874547 0.13337693 0.28021662 0.17925207 0.60122274 0.25502619
##        289        290        291        292        293        294
## 0.23197657 0.05878643 0.38047126 0.35008696 0.46240272 0.73372225
##        295        296        297        298        299        300
## 0.29885443 0.17659931 0.45483793 0.23950580 0.34785059 0.27566478
##        301        302        303        304        305        306
## 0.36288468 0.28067279 0.22671860 0.51860565 0.07198547 0.19060160
##        307        308        309        310        311        312
## 0.44561844 0.37054412 0.28373804 0.12588934 0.30028221 0.44520022
##        313        314        315        316        317        318
## 0.30907647 0.19322270 0.17701800 0.15412239 0.18491373 0.29806393
##        319        320        321        322        323        324
## 0.18670880 0.46755914 0.14630641 0.32183935 0.12035456 0.17486941
##        325        326        327        328        329        330
## 0.12112920 0.66498227 0.38597852 0.35450549 0.33926538 0.11370930
##        331        332        333        334        335        336
## 0.39213236 0.27905234 0.34097123 0.21344965 0.20393972 0.59795326
##        337        338        339        340        341        342
## 0.16520993 0.16070084 0.45158492 0.26006097 0.14037382 0.12659514
##        343        344        345        346        347        348
## 0.22560760 0.29075910 0.18859648 0.14657301 0.35132030 0.42636137
##        349        350        351        352        353        354
## 0.25767548 0.27488628 0.57858815 0.23714608 0.18120291 0.43779599
##        355        356        357        358        359        360
## 0.40050290 0.49758253 0.38909423 0.57487559 0.25063922 0.37007654
##        361        362        363        364        365        366
## 0.59956970 0.50972425 0.35412991 0.29777892 0.49491656 0.11836196
##        367        368        369        370        371        372
## 0.12645014 0.26745319 0.63170496 0.56803162 0.39857395 0.31708679
##        373        374        375        376        377        378
## 0.37650752 0.53085361 0.41142403 0.18735742 0.41512421 0.58958954
##        379        380        381        382        383        384
## 0.20223990 0.21896113 0.46366743 0.34602886 0.34967678 0.67275941
##        385        386        387        388        389        390
## 0.18665107 0.35189341 0.52842881 0.34287938 0.33908140 0.40275050
##        391        392        393        394        395        396
## 0.40093595 0.48719398 0.22202911 0.43872524 0.25342327 0.48866999
##        397        398        399        400
## 0.16550430 0.18106222 0.46366743 0.30073055
```

9) Create another vector admit.pred to show 0 or 1 for admit.prob. Let's set the value to be 0 if the probability is less than 0.5, and 1 if the probability is no less than 0.5.

```
admit.pred<-rep(0,400)
admit.pred[admit.probs>.5] <- 1
```

10) Using table() function to create a confusion matrix to determines how many observations were correctly or incorrectly classified. Calculate the percentage that the observations were correctly classified.

```
table(admit.pred,mydata$admit)
```

```
##
## admit.pred   0   1
##          0 254  97
##          1  19  30
```

```
mean(admit.pred == mydata$admit)
```

```
## [1] 0.71
```

Correctly classified = (254+30)/400 = 71%

11) Use the model to predict the average cases in each rank, that is, four new data with mean gre, mean gpa and rank from 1 to 4.

```
newdata1 <- with(mydata, data.frame(gre = mean(gre), gpa = mean(gpa), rank = factor(1:4)))
newdata1$admit1.prob <- predict(glm.fit, newdata = newdata1, type = "response")
newdata1
```

```
##     gre    gpa rank admit1.prob
## 1 587.7 3.3899    1   0.5166016
## 2 587.7 3.3899    2   0.3522846
## 3 587.7 3.3899    3   0.2186120
## 4 587.7 3.3899    4   0.1846684
```

```
newdata1$admit1.pred <- rep(1,4)
newdata1$admit1.pred[newdata1$admit1.prob<0.5] <- 0
newdata1
```

```
##     gre    gpa rank admit1.prob admit1.pred
## 1 587.7 3.3899    1   0.5166016           1
## 2 587.7 3.3899    2   0.3522846           0
## 3 587.7 3.3899    3   0.2186120           0
## 4 587.7 3.3899    4   0.1846684           0
```