Chapter 18 - Logistic Regression

**Binomial Logistic Regression**

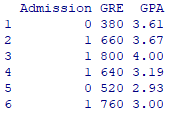
Admission.csv contains the data of 400 students’ GRE and GPA grades, and whether or not they were admitted to a university. Perform a binomial logistic regression to see if GRE and GPE grades are significant predictors of being admitted to a university.

Ensure that the levels are set appropriately, interpret the results and analyze the accuracy of the model. The answers only refer to the whole file, so for comparison it is recommended that you strip out references to 'train' and 'test'.

When doing this, references from the book to 'train' and 'test' have been stripped out.

file = read.csv("Admission.csv")

head(file)



We need to make sure that Admission has the correct levels, but it seems possible that 0 and 1 may not be factors. Let's check:

levels(file$Admission)



class(file$Admission)



We need to turn Admission into a factor:

library(car)

file$Admission = Recode(file$Admission, " ", as.factor = TRUE)

The speech marks surrounding a gap mean that individual elements do not require changes. (See Chapter 21, Data handling section, Recoding existing data subsection, for more details.)

class(file$Admission)

 # That's more like it

levels(file$Admission)



Let us say that we are more interested in a student being admitted than not. That being the case, '0' may as well remain as the baseline, so we don't need to relevel.

Note that there is no 'Admission = file$Admission' short-cut. We want to handle the whole 'file' object in the later procedures.

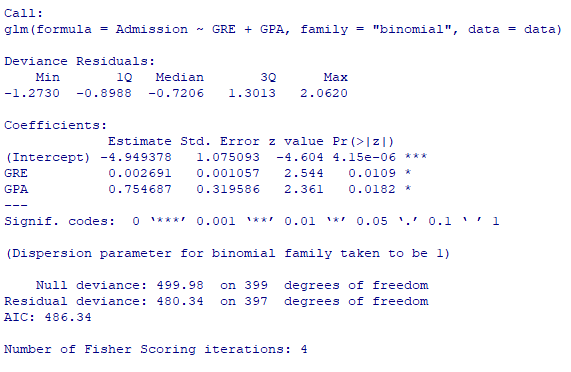
Fitting the model:

data = file

model1 = glm(Admission ~ GRE + GPA,

data, family="binomial")

summary(model1)

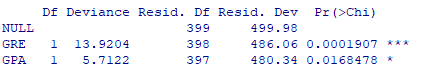


The estimates for GRE and GPA are both in the expected positive direction, meaning that admissions are more likely with higher GRE and GPA results. The *z* values and *p* values indicate significant results for both variables. .

confint(model1)



anova(model1, test="Chisq")



The deviance and AIC values assess the model fit. AIC is more useful when comparing two or more models with different predictors. In terms of magnitude, the smaller the AIC, the better. Using fewer variables (removing the insignificant ones) often leads to a larger AIC as a trade-off. Thus, it is necessary to observe if the difference in AIC is large or negligible.

In this case, we are only considering one model since both predictors turned out to be significant (see model coefficients table). Now let's look at the odds ratios:

exp(coef(model1))



For a one-unit increase in GRE, the odds of a student being admitted are increased by a factor of 1.003. For a one-unit increase in GPA, the odds of a student being admitted are increased by a factor of 2.127. Let's look at confidence intervals for the odds ratios:

exp(confint(model1))



library(pscl)

round(pR2(model1), 3)



Pseudo R-squared statistics. These are often quite small.

**Prediction**

When doing this, strip out references in the book to 'train' and 'test'. Just use 'file'.

newtest = with(file, data.frame(

GRE = mean(GRE), GPA = mean(GPA))

newtest$Prob = predict(

model1, newdata = newtest, type = "response")

Classification = ifelse(newtest$Prob > 0.5,'1','0')

newtest$Admission=

ifelse(file$Admission == 1,1,0)

Accuracy = mean(Classification == newtest$Admission)

Accuracy



The overall accuracy of the model is 68%.

dv = file$Admission # The outcome variable of the test sample

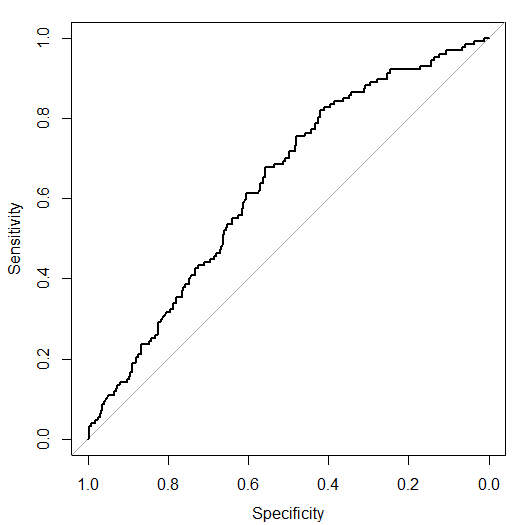
prediction = predict(

model1, newdata=file, type=c("response"))

library(pROC)

rocCurve = roc(dv ~ prediction)

plot(rocCurve)



auc(rocCurve)



The area under the ROC curve is 0.63 which falls under the 'poor' category. Note that there are only 2 predictors in this model. The amount of variance that cannot be accounted for by these two predictors is still large, causing low predictive power.

ci.auc(rocCurve)



Even worse!

Specificity and sensitivity:

ci.thresholds(rocCurve, thresholds=0.5)



|  |
| --- |
|  |

The test is highly specific, meaning that it is better at predicting those who will not be admitted than those will get admitted. This is still useful in a sense, in that, if a student is predicted to be admitted, realistically, there is a high chance that the student really will be admitted to university.

The cut-off prediction, 'Classification', was set to 0.5. You can experiment with the cut off and try to prioritize correct classification for either admitted students or non-admitted students.

Effects charts

model1a = glm(Admission ~ GRE + GPA,

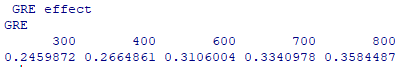
data = test, family="binomial")

library(effects)

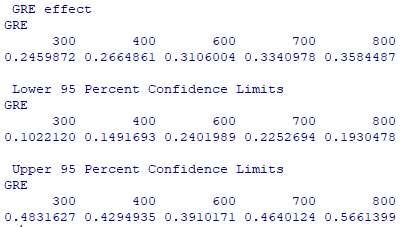
plot(effect("GRE", model1a), style="lines")



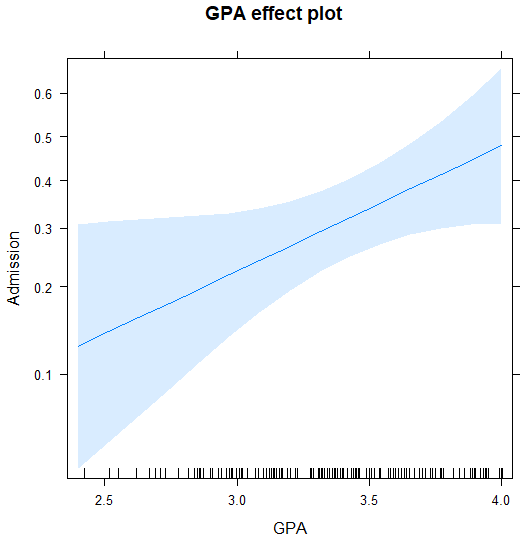
effect("GRE", model1a) # Gives precise means



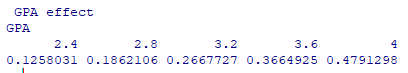
summary(effect("GRE", model1a)) # With confidence limits



plot(effect("GPA", model1a), style="lines")



effect("GPA", model1a) # Gives precise means



summary(effect("GPA", model1a)) # With confidence limits

