Module7\_Homework.R

mmhan\_uricwmy

2022-08-08

# =====================================================  
# Module 7 Homework - Logistic Regression  
#   
# Mike Hankinson  
# November 12, 2021  
# =====================================================  
  
# Given:   
# - Assignment7.csv: Student GRE scores and GPA and a binary column indicating whether they   
# were admitted to a certain university.   
  
# Pre-processing:  
# a. Load in the data.   
# b. GRE and GPA are measure on significantly different scales.   
# To allow us to interpret these variables on the same range, scale both variables using   
# standardization. This means each variable will have a mean of 0 and a standard   
# deviation of 1.   
# c. Set the dependent variable "admit" as a factor variable and perform logistic regression   
# with two predictors: GRE and GPA.   
  
# Questions:   
# 7 questions defined and answered throughout the code.   
  
  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# Pre-processing   
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
# a. Load and Plot Data

# Data of 400 Students  
dat <- read.csv("Assignment7.csv")  
names(dat) # [1] "admit" "GRE" "GPA"

## [1] "admit" "GRE" "GPA"

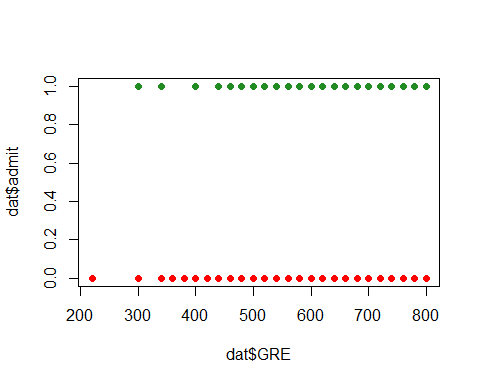
head(dat)

## admit GRE GPA  
## 1 0 380 3.61  
## 2 1 660 3.67  
## 3 1 800 4.00  
## 4 1 640 3.19  
## 5 0 520 2.93  
## 6 1 760 3.00

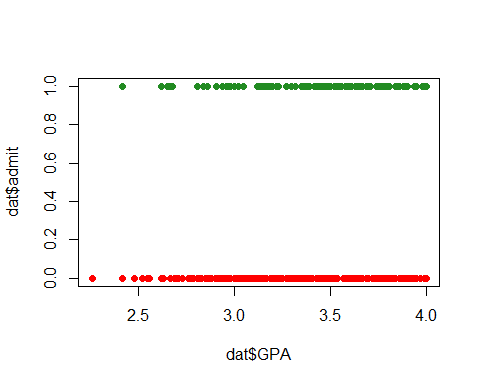
tail(dat)

## admit GRE GPA  
## 395 1 460 3.99  
## 396 0 620 4.00  
## 397 0 560 3.04  
## 398 0 460 2.63  
## 399 0 700 3.65  
## 400 0 600 3.89

my.color <- c("red", "forestgreen")[dat$admit+1]  
plot(dat$GRE,dat$admit, pch=16, col=my.color)



my.color <- c("red", "forestgreen")[dat$admit+1]  
plot(dat$GPA,dat$admit, pch=16, col=my.color)



# b. Scale Both Variables Using Standardization

# https://stackoverflow.com/questions/8120984/scaling-data-in-r-ignoring-specific-columns  
  
dat.scale <- dat  
dat.scale[, -c(1)] <- scale(dat.scale[, -c(1)])  
head(dat.scale)

## admit GRE GPA  
## 1 0 -1.7980110 0.5783479  
## 2 1 0.6258844 0.7360075  
## 3 1 1.8378321 1.6031352  
## 4 1 0.4527490 -0.5252692  
## 5 0 -0.5860633 -1.2084607  
## 6 1 1.4915613 -1.0245245

# admit GRE GPA  
 # 1 0 -1.7980110 0.5783479  
 # 2 1 0.6258844 0.7360075  
 # 3 1 1.8378321 1.6031352  
 # 4 1 0.4527490 -0.5252692  
 # 5 0 -0.5860633 -1.2084607  
 # 6 1 1.4915613 -1.0245245  
  
# Verify mean and standard deviations of GRE and GPA post-scaling  
sd\_GRE <- sd(dat.scale$GRE) # [1] 1  
sd\_GPA <- sd(dat.scale$GPA) # [1] 1  
mgre1 <- mean(dat.scale$GRE) # 0  
mgpa2 <- mean(dat.scale$GPA) # 0  
  
  
  
# c. Convert Dependent Variable to Type Factor / Run Logistic Regression Model

# Since the dependent variable is categorical and not numeric,   
# convert it to type factor for logistic regression and subsequent analysis.  
Response <- as.factor(dat.scale$admit)  
 # Levels: 0 1  
  
# Run Logistic Regression Model  
logistic.regression <- glm(Response ~ GRE + GPA, family="binomial", data=dat.scale)  
summary(logistic.regression)

##   
## Call:  
## glm(formula = Response ~ GRE + GPA, family = "binomial", data = dat.scale)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.2730 -0.8988 -0.7206 1.3013 2.0620   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.8098 0.1120 -7.233 4.74e-13 \*\*\*  
## GRE 0.3108 0.1222 2.544 0.0109 \*   
## GPA 0.2872 0.1216 2.361 0.0182 \*   
## ---  
## 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: 480.34 on 397 degrees of freedom  
## AIC: 486.34  
##   
## Number of Fisher Scoring iterations: 4

# Deviance Residuals:   
 # Min 1Q Median 3Q Max   
 # -1.2730 -0.8988 -0.7206 1.3013 2.0620   
 #   
 # Coefficients:  
 # Estimate Std. Error z value Pr(>|z|)   
 # (Intercept) -0.8098 0.1120 -7.233 4.74e-13 \*\*\*  
 # GRE 0.3108 0.1222 2.544 0.0109 \*   
 # GPA 0.2872 0.1216 2.361 0.0182 \*   
 # ---  
 # 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: 480.34 on 397 degrees of freedom  
 # AIC: 486.34  
 #   
 # Number of Fisher Scoring iterations: 4  
  
  
# LR Model Conclusions:   
 # 1. GRE B0 = 0.311 states for every additional point earned in the GRE test   
 # the probability of acceptance to the university increases by that amount.   
 # 2. GPA B0 = 0.287 states for every additional point added to a student's GPA  
 # the probability of acceptance to the university increases by that amount.   
  
  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
# Questions  
# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
# 1.Provide an interpretation for the intercept coefficient.   
# What does it mean if both predictors are equal to 0?

#Note:The dependent variable (admit) of the original data set is a probability  
# bound by 0 and 1. The logistic regression model transforms admit to a  
# continuous variable to match both GPA and GRE via the log of the odds  
# -- or, log of p(admit)/(1-p(admit)) then computes by minimizing the sum of   
# the logistic loss.  
  
# - The intercept coefficient, B0=-0.8098.   
# - This is the log odds of acceptance to the university when both predictors = 0.   
# - A negative log odds means that the odds of acceptance is less than 0.50  
# - Taking e to the log odds presents an easier view of the number and converts to odds   
Odds <- exp(logistic.regression$coefficients[1]) # 0.4449691   
  
# Perform Verification  
B.coefficients <- logistic.regression$coefficients  
 # (Intercept) GRE GPA   
 # -0.8097503 0.3108184 0.2872087   
lodds0 <- as.numeric(c(B.coefficients[1]+B.coefficients[2]\*0+B.coefficients[3]\*0))  
 # [1] -0.8097503, this matches the glm model output for B0 (y-intercept).   
  
  
  
prob0 <- 1/(1+exp(-lodds0))  
probablity.at.intercept <- Odds/(1-Odds)  
  
  
  
# 2a. Assuming an average value for GRE, calculate the effect of a one unit   
# increase around the mean for GPA.

# Marginal Effect at the Mean (EAM) is a way to calculate the probability   
# increase given a one unit increase around the mean of the independent variable.   
  
  
mgre <- mean(dat$GRE) # [1] 587.7  
mgre1 # [1] -4.010984e-16 or 0

## [1] -4.010984e-16

mgpa <- mean(dat$GPA) # [1] 3.3899  
mgpa2 # [1] 2.272705e-16 or 0

## [1] 2.272705e-16

meanGPA <- data.frame(GPA=c(mgpa2-1, mgpa2+1), GRE=mgre1)  
  
# With type="response", the function returns PROBABILITIES   
pmean1 <- predict(logistic.regression, newdata=meanGPA, type="response")  
(EAM1 <- pmean1[2]-pmean1[1])

## 2   
## 0.121948

# Conclusion:   
# 0.121948 is the increase in probability of admission, maintaining a constant average GRE,   
# given a one unit increase in GPA.  
  
  
  
# 2b. Assuming an average value for GPA, calculate the effect of a one unit   
# increase around the mean for GRE.

meanGRE <- data.frame(GRE=c(mgre1-1, mgre1+1), GPA=mgpa2)  
pmean2 <- predict(logistic.regression, newdata=meanGRE, type="response")  
(EAM1 <- pmean2[2]-pmean2[1])

## 2   
## 0.1318859

# Conclusion:   
# 0.1318859 is the increase in probability of admission, maintaining a constant average GPA,   
# given a one unit increase in GRE score.  
  
  
  
# 3a. With an average value for GRE, calculate the probability of being admitted   
# under the following conditions for GPA: 3 SD below mean, 2.5 SD below mean,   
# 2 SD below mean, 1.5 SD below mean, 1 SD below mean, 0.5 SD below mean, Mean score,   
# 0.5 SD above mean, 1 SD above mean, 1.5 SD above mean, 2 SD above mean.   
# What is the average marginal effect?

GPA\_prob\_df <- data.frame(GRE=mgre1, GPA=c(mgpa2-3\*sd\_GPA, mgpa2-2.5\*sd\_GPA, mgpa2-2\*sd\_GPA, mgpa2-1.5\*sd\_GPA,   
 mgpa2-1\*sd\_GPA, mgpa2-0.5\*sd\_GPA, mgpa2-0\*sd\_GPA, mgpa2+0.5\*sd\_GPA,  
 mgpa2+1\*sd\_GPA, mgpa2+1.5\*sd\_GPA, mgpa2+2\*sd\_GPA))  
  
GPA\_probabilities <- predict(logistic.regression, newdata=GPA\_prob\_df, type="response")  
GPA\_ROw\_Titles <- c("-3SD", "-2.5SD", "-2SD", "-1.5SD", "-1SD", "-0.5 SD", "SD", "+0.5SD", "+1SD", "+1.5SD",  
 "+2SD")  
  
  
# Conclusion: The probability of being admitted increases with increasing positive  
# sd from the mean of the GPA (with constant GRE at the mean):  
GPA\_probabilities\_summary <- cbind(GPA\_ROw\_Titles, GPA\_probabilities)  
 # GPA\_ROw\_Titles GPA\_probabilities   
 # 1 "-3SD" "0.158240733769537"  
 # 2 "-2.5SD" "0.178319874668119"  
 # 3 "-2SD" "0.200340463497387"  
 # 4 "-1.5SD" "0.224337924193125"  
 # 5 "-1SD" "0.250310104606916"  
 # 6 "-0.5 SD" "0.278210663318401"  
 # 7 "SD" "0.307943699230024"  
 # 8 "+0.5SD" "0.339360358993962"  
 # 9 "+1SD" "0.372258114944254"  
 # 10 "+1.5SD" "0.40638324989128"   
 # 11 "+2SD" "0.441436812674739"  
  
  
all.effects <- diff(GPA\_probabilities)  
AME <- mean(all.effects) # [1] 0.02831961  
# Conclusion: The marginal effect of GPA = 0.02831961, keeping GRE constant at its mean.   
  
  
  
# 3b. With an average value for GPA, calculate the probability of being admitting under   
# the following conditions for GRE: 3 SD below mean, 2.5 SD below mean, 2 SD below mean,   
# 1.5 SD below mean, 1 SD below mean, 0.5 SD below mean, Mean score, 0.5 SD above mean,   
# 1 SD above mean, 1.5 SD above mean, 2 SD above mean.   
# What is the average marginal effect?

GRE\_prob\_df <- data.frame(GPA=mgpa2, GRE=c(mgre1-3\*sd\_GRE, mgre1-2.5\*sd\_GRE, mgre1-2\*sd\_GRE, mgre1-1.5\*sd\_GRE,   
 mgre1-1\*sd\_GRE, mgre1-0.5\*sd\_GRE, mgre1-0\*sd\_GRE, mgre1+0.5\*sd\_GRE,  
 mgre1+1\*sd\_GRE, mgre1+1.5\*sd\_GRE, mgre1+2\*sd\_GRE))  
  
GRE\_probabilities <- predict(logistic.regression, newdata=GRE\_prob\_df, type="response")  
GRE\_ROw\_Titles <- c("-3SD", "-2.5SD", "-2SD", "-1.5SD", "-1SD", "-0.5 SD", "SD", "+0.5SD", "+1SD", "+1.5SD",  
 "+2SD")  
  
  
# Conclusion: The probability of being admitted increases with increasing positive  
# sd from the mean of GRE score (with constant GPA at the mean):  
GRE\_probabilities\_summary <- cbind(GRE\_ROw\_Titles, GRE\_probabilities)  
 # GRE\_ROw\_Titles GRE\_probabilities   
 # 1 "-3SD" "0.149032987735846"  
 # 2 "-2.5SD" "0.169835089498502"  
 # 3 "-2SD" "0.192882628317169"  
 # 4 "-1.5SD" "0.218235632202989"  
 # 5 "-1SD" "0.245905794139975"  
 # 6 "-0.5 SD" "0.275846354162737"  
 # 7 "SD" "0.307943699230024"  
 # 8 "+0.5SD" "0.342011944731309"  
 # 9 "+1SD" "0.377791711245609"  
 # 10 "+1.5SD" "0.414954033958035"  
 # 11 "+2SD" "0.453109832091711"  
  
  
all.effects2 <- diff(GRE\_probabilities)  
AME2 <- mean(all.effects2) # [1] 0.03040768  
# Conclusion: The marginal effect of GRE = 0.03040768, keeping GPA constant at its mean.   
  
  
  
  
# 4. How many standard deviations above the mean should your GRE score be if your GPA is  
# 0.5 standard deviations below the mean and you'd like a 75% chance of being admitted?

# Givens:   
# 1. P = 0.75 that y=1  
# 2. GPA = mgpa2 - 0.5\*sd\_GPA  
  
# Unknowns:  
# 1. x = # of sd above sd\_GRE; GRE = mgre1 + x\*sd\_GRE  
  
# Solution:  
p <- 0.75  
GPA.prob4 = mgpa2-0.5\*sd\_GPA # [1] -0.5   
log.odds4 = log(p/(1-p)) # [1] 1.098612  
B.coefficients[3]

## GPA   
## 0.2872087

X1 <- (log.odds4-B.coefficients[3]\*GPA.prob4-B.coefficients[1])/B.coefficients[2]  
 # 7.063839 is the value for X1 (this seems way too high, imo)  
# To find the number of SDs above the mean,   
GRE\_SD\_above\_mean <- (X1 - mgre1)/sd\_GRE # (7.06-0)/1  
  
# Therefore, must be 7.1 standard deviations above mean GRE score to have a 75% probability  
# of admittance with a GPA 0.5 standard deviations below its mean.   
  
  
# 5. Multiply the intercept by -1. Divide this value by the sum of the two slope coefficients.   
# Use this result as values for an observation of GRE and GPA and calculate the output from   
# the model. What's your interpretation?

GRE.GPA.Value <- (logistic.regression$coefficients[1]\*-1)/(logistic.regression$coefficients[2]+  
 logistic.regression$coefficients[3])  
 # (Intercept)   
 # 1.354036   
  
log.odds5 <- logistic.regression$coefficients[1]+logistic.regression$coefficients[2]\*GRE.GPA.Value+  
 logistic.regression$coefficients[3]\*GRE.GPA.Value  
 # -5.551115e-17   
exp(log.odds5)

## (Intercept)   
## 1

p5 <- exp(log.odds5)/2  
 # (Intercept)   
 # 0.5  
  
# Conclusion:  
# logodds = 0  
# logodds = ln(p(y)/(1-py)) = 0  
# Taking e^ to both sides yields, 1 = p(y)/(1-py)  
# Probability of acceptance at these conditions is 50%.   
  
  
  
# 6. Generate predictions using a 50% classification boundary.   
# Report overall accuracy and balance accuracy.   
# Feel free to share any other metrics you find interesting.   
# Are you satisfied with this classification boundary?   
# If yes, say why. If not, evaluate results when using another classification boundary.

# Define Log of Odds >= 0.5 as pass, 1  
# Log of Odds < 0.5 as fail, 0  
  
predictions <- logistic.regression$fitted.values  
predictions[predictions>=0.5] <- 1  
predictions[predictions<0.5] <- 0  
predictions <- as.factor(predictions)  
table(predictions, dat$admit)

##   
## predictions 0 1  
## 0 263 118  
## 1 10 9

# predictions 0 1  
 # 0 263 118  
 # 1 10 9  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

confusionMatrix(predictions, Response, mode="prec\_recall", positive = "1")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction 0 1  
## 0 263 118  
## 1 10 9  
##   
## Accuracy : 0.68   
## 95% CI : (0.6318, 0.7255)  
## No Information Rate : 0.6825   
## P-Value [Acc > NIR] : 0.5665   
##   
## Kappa : 0.0443   
##   
## Mcnemar's Test P-Value : <2e-16   
##   
## Precision : 0.47368   
## Recall : 0.07087   
## F1 : 0.12329   
## Prevalence : 0.31750   
## Detection Rate : 0.02250   
## Detection Prevalence : 0.04750   
## Balanced Accuracy : 0.51712   
##   
## 'Positive' Class : 1   
##

# Accuracy : 0.68   
 # 95% CI : (0.6318, 0.7255)  
 # No Information Rate : 0.6825   
 # P-Value [Acc > NIR] : 0.5665   
 #   
 # Kappa : 0.0443   
 #   
 # Mcnemar's Test P-Value : <2e-16   
 #   
 # Precision : 0.47368   
 # Recall : 0.07087   
 # F1 : 0.12329   
 # Prevalence : 0.31750   
 # Detection Rate : 0.02250   
 # Detection Prevalence : 0.04750   
 # Balanced Accuracy : 0.51712   
  
  
# 7. Plot an ROC curve and report the area under the curve.   
# Based on this and your classification predictions, how do you evaluate the ability of this   
# model to use GRE score and GPA to differentiate between whether students will be admitted?

library(AUC)

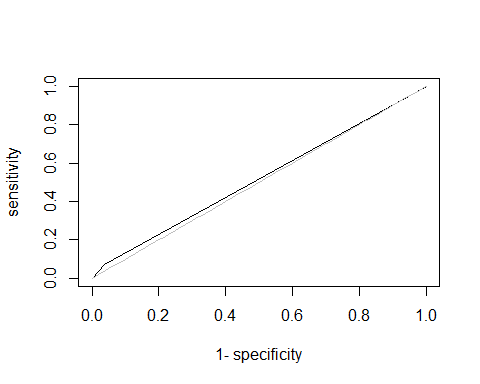
## AUC 0.3.0

## Type AUCNews() to see the change log and ?AUC to get an overview.

##   
## Attaching package: 'AUC'

## The following objects are masked from 'package:caret':  
##   
## sensitivity, specificity

r <- roc(predictions, Response) # Remember: Since the dependent variable is   
# categorical and not numeric,   
# converted it to type factor above for   
# logistic regression and subsequent analysis by  
# Response <- as.factor(dat$Pass)  
  
  
plot(roc(predictions, Response))



auc(r) # [1] 0.5171181

## [1] 0.5171181

# The area under the ROC curve can be computed to suggest how useful our model   
# is for distinguishing these two classes. The following scale can be used to   
# interpret area under the curve:  
# 0.517: Random model, no ability to distinguish  
  
  
  
  
  
  
  
  
  
# rm(list = ls()) Removes global environment