Homework 3

4375 Machine Learning with Dr. Mazidi

Tyler Echols

6/13/2022

This homework runs logistic regression to predict the binary feature of whether or not a person was admitted to graduate school, based on a set of predictors: GRE score, TOEFL score, rating of undergrad university attended, SOP statement of purpose, LOR letter or recommendation, Undergrad GPA, Research experience (binary).

The data set was downloaded from Kaggle: <https://www.kaggle.com/mohansacharya/graduate-admissions>

The data is available in Piazza.

## Step 1 Load the data

* Load the data
* Examine the first few rows with head()

# your code here  
df <- read.csv("Admission\_Predict.csv", header = TRUE)  
head(df)

## Serial.No. GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research  
## 1 1 337 118 4 4.5 4.5 9.65 1  
## 2 2 324 107 4 4.0 4.5 8.87 1  
## 3 3 316 104 3 3.0 3.5 8.00 1  
## 4 4 322 110 3 3.5 2.5 8.67 1  
## 5 5 314 103 2 2.0 3.0 8.21 0  
## 6 6 330 115 5 4.5 3.0 9.34 1  
## Chance.of.Admit  
## 1 0.92  
## 2 0.76  
## 3 0.72  
## 4 0.80  
## 5 0.65  
## 6 0.90

## Step 2 Data Wrangling

Perform the following steps:

* Make Research a factor
* Get rid of the Serial No column
* Make a new column that is binary factor based on if Chance.of.Admit > 0.5. Hint: See p. 40 in the book.
* Output column names with names() function
* Output a summary of the data
* Is the data set unbalanced? Why or why not?

Your commentary here: It looks balanced because on each graph because it seems you to be sharing the same information.

# your code here  
df$Research <- factor(df$Research)  
df$Serial.No. <- NULL  
head(df)

## GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research Chance.of.Admit  
## 1 337 118 4 4.5 4.5 9.65 1 0.92  
## 2 324 107 4 4.0 4.5 8.87 1 0.76  
## 3 316 104 3 3.0 3.5 8.00 1 0.72  
## 4 322 110 3 3.5 2.5 8.67 1 0.80  
## 5 314 103 2 2.0 3.0 8.21 0 0.65  
## 6 330 115 5 4.5 3.0 9.34 1 0.90

df$bFactor <- ifelse(df$Chance.of.Admit > 0.5, 1, 0)  
head(df)

## GRE.Score TOEFL.Score University.Rating SOP LOR CGPA Research Chance.of.Admit  
## 1 337 118 4 4.5 4.5 9.65 1 0.92  
## 2 324 107 4 4.0 4.5 8.87 1 0.76  
## 3 316 104 3 3.0 3.5 8.00 1 0.72  
## 4 322 110 3 3.5 2.5 8.67 1 0.80  
## 5 314 103 2 2.0 3.0 8.21 0 0.65  
## 6 330 115 5 4.5 3.0 9.34 1 0.90  
## bFactor  
## 1 1  
## 2 1  
## 3 1  
## 4 1  
## 5 1  
## 6 1

names(df)

## [1] "GRE.Score" "TOEFL.Score" "University.Rating"  
## [4] "SOP" "LOR" "CGPA"   
## [7] "Research" "Chance.of.Admit" "bFactor"

# put the summary here  
summary(df)

## GRE.Score TOEFL.Score University.Rating SOP   
## Min. :290.0 Min. : 92.0 Min. :1.000 Min. :1.0   
## 1st Qu.:308.0 1st Qu.:103.0 1st Qu.:2.000 1st Qu.:2.5   
## Median :317.0 Median :107.0 Median :3.000 Median :3.5   
## Mean :316.8 Mean :107.4 Mean :3.087 Mean :3.4   
## 3rd Qu.:325.0 3rd Qu.:112.0 3rd Qu.:4.000 3rd Qu.:4.0   
## Max. :340.0 Max. :120.0 Max. :5.000 Max. :5.0   
## LOR CGPA Research Chance.of.Admit bFactor   
## Min. :1.000 Min. :6.800 0:181 Min. :0.3400 Min. :0.0000   
## 1st Qu.:3.000 1st Qu.:8.170 1:219 1st Qu.:0.6400 1st Qu.:1.0000   
## Median :3.500 Median :8.610 Median :0.7300 Median :1.0000   
## Mean :3.453 Mean :8.599 Mean :0.7244 Mean :0.9125   
## 3rd Qu.:4.000 3rd Qu.:9.062 3rd Qu.:0.8300 3rd Qu.:1.0000   
## Max. :5.000 Max. :9.920 Max. :0.9700 Max. :1.0000

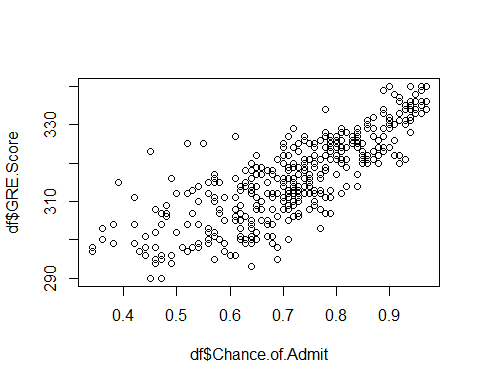
## Step 3 Data Visualization

* Create a side-by-side graph with Admit on the x axis of both graphs, GRE score on the y axis of one graph and TOEFL score on the y axis of the other graph; save/restore the original graph parameters
* Comment on the graphs and what they are telling you about whether GRE and TOEFL are good predictors
* You will get a lot of warnings, you can suppress them with disabling warnings as shown below:

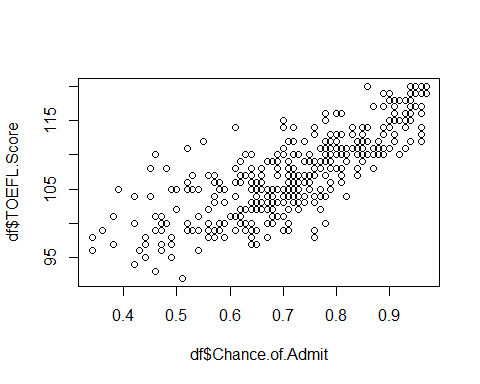
{r,warning=FALSE}

Your commentary here: Both graph’s are going up at an gradualling increasing pace.

# your code here  
plot(df$Chance.of.Admit, df$GRE.Score)



plot(df$Chance.of.Admit, df$TOEFL.Score)



## Step 4 Divide train/test

* Divide into 75/25 train/test, using seed 1234

# your code here  
set.seed(1234)  
sample <- sample.int(n=nrow(df), size=floor(.75\*nrow(df)), replace = F)  
train = df[sample,]  
test = df[-sample,]

## Step 5 Build a Model with all predictors

* Build a model, predicting Admit from all predictors
* Output a summary of the model
* Did you get an error? Why? Hint: see p. 120 Warning

Your commentary here: the error are got are talking about how they can not fit between the numerically set values

# your code here  
glm1 <- glm(bFactor~., family=binomial,data=train)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

summary(glm1)

##   
## Call:  
## glm(formula = bFactor ~ ., family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -9.801e-05 2.100e-08 2.100e-08 2.100e-08 1.123e-04   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -6.465e+02 2.921e+05 -0.002 0.998  
## GRE.Score -3.617e-01 9.554e+02 0.000 1.000  
## TOEFL.Score 3.551e+00 3.562e+03 0.001 0.999  
## University.Rating -5.000e+00 1.511e+04 0.000 1.000  
## SOP -7.867e+00 1.262e+04 -0.001 1.000  
## LOR -4.673e+00 1.970e+04 0.000 1.000  
## CGPA 3.605e+00 1.897e+04 0.000 1.000  
## Research1 -1.109e+01 1.199e+04 -0.001 0.999  
## Chance.of.Admit 7.993e+02 1.610e+05 0.005 0.996  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1.7685e+02 on 299 degrees of freedom  
## Residual deviance: 5.7812e-08 on 291 degrees of freedom  
## AIC: 18  
##   
## Number of Fisher Scoring iterations: 25

## Step 6 Build a Model with all predictors except Chance.of.Admit

* Build another model, predicting Admit from all predictors *except* Chance.of.Admit
* Output a summary of the model
* Did you get an error? Why or why not? # There were no error’s for this because It has a more defined bound to work with

# your code here  
glm2 <- glm(bFactor~. -Chance.of.Admit , family=binomial,data=train)  
summary(glm2)

##   
## Call:  
## glm(formula = bFactor ~ . - Chance.of.Admit, family = binomial,   
## data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.98738 0.02404 0.08347 0.25965 1.79020   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -52.42714 12.25908 -4.277 1.9e-05 \*\*\*  
## GRE.Score 0.01685 0.04566 0.369 0.712200   
## TOEFL.Score 0.17305 0.10614 1.630 0.103027   
## University.Rating -0.66933 0.40166 -1.666 0.095631 .   
## SOP -0.81828 0.45026 -1.817 0.069161 .   
## LOR 1.22762 0.54752 2.242 0.024951 \*   
## CGPA 3.94613 1.07273 3.679 0.000235 \*\*\*  
## Research1 0.10073 0.73916 0.136 0.891600   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 176.854 on 299 degrees of freedom  
## Residual deviance: 89.024 on 292 degrees of freedom  
## AIC: 105.02  
##   
## Number of Fisher Scoring iterations: 8

## Step 7 Predict probabilities

* Predict the probabilities using type=“response”
* Examine a few probabilities and the corresponding Chance.of.Admit values
* Run cor() on the predicted probs and the Chance.of.Admit, and output the correlation
* What do you conclude from this correlation.

Your commentary here:that these probabilities are still in the middle of 0 and 1.

# your code here  
probs <- predict(glm2, newdata=df, type="response")  
head(probs)

## 1 2 3 4 5 6   
## 0.9999835 0.9980217 0.9165608 0.9894708 0.9779368 0.9986996

cor(probs,df$Chance.of.Admit)

## [1] 0.6338116

## Step 8 Make binary predictions, print table and accuracy

* Now make binary predictions
* Output a table comparing the predictions and the binary Admit column
* Calculate and output accuracy
* Was the model able to generalize well to new data?

Your commentary here:The model only repeated new data that was already presented.

# your code here  
glm2 <- glm(bFactor~., family=binomial,data=train)

## Warning: glm.fit: algorithm did not converge

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

probs <- predict(glm2, newdata=test, type="response")  
pred <- ifelse(probs> 0.5, 2, 1)  
acc <- mean(pred == as.integer(test$bFactor))  
summary(glm2)

##   
## Call:  
## glm(formula = bFactor ~ ., family = binomial, data = train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -9.801e-05 2.100e-08 2.100e-08 2.100e-08 1.123e-04   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -6.465e+02 2.921e+05 -0.002 0.998  
## GRE.Score -3.617e-01 9.554e+02 0.000 1.000  
## TOEFL.Score 3.551e+00 3.562e+03 0.001 0.999  
## University.Rating -5.000e+00 1.511e+04 0.000 1.000  
## SOP -7.867e+00 1.262e+04 -0.001 1.000  
## LOR -4.673e+00 1.970e+04 0.000 1.000  
## CGPA 3.605e+00 1.897e+04 0.000 1.000  
## Research1 -1.109e+01 1.199e+04 -0.001 0.999  
## Chance.of.Admit 7.993e+02 1.610e+05 0.005 0.996  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 1.7685e+02 on 299 degrees of freedom  
## Residual deviance: 5.7812e-08 on 291 degrees of freedom  
## AIC: 18  
##   
## Number of Fisher Scoring iterations: 25

## Step 9 Output ROCR and AUC

* Output a ROCR graph
* Extract and output the AUC metric

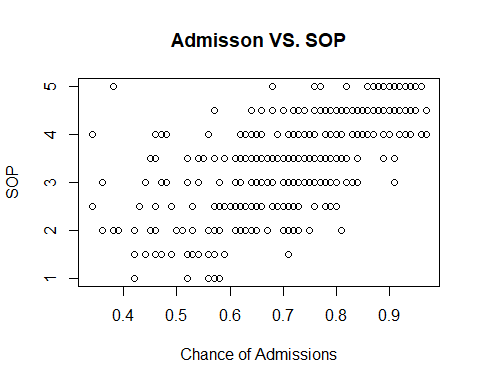
# your code here  
##library(ROCR)  
##rocNew <- roc(df$Chance.of.Admit, glm2$fitted.values, plot = TRUE)  
##rocNew  
##cat("Area under the curve: ", rocNew$auc)

## Step 10

* Make two more graphs and comment on what you learned from each graph:
  + Admit on x axis, SOP on y axis
  + Research on x axis, SOP on y axis

Your commentary here: Both graphic have very low predictors and and random variables.

# plot 1  
plot( x = df$Chance.of.Admit, y = df$SOP, main = "Admisson VS. SOP", ylab = "SOP", xlab = "Chance of Admissions")



# plot 2  
plot (x = df$Research, y = df$SOP, main = "Research and SOP", ylab = "SOP", xlab = "Research")

