Homework 3

4375 Machine Learning with Dr. Mazidi

Cris Chou

9/14

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")
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: The data set is unbalanced because a disproportionate percent of the dataset had a higher than .5 chance of getting admitted. In a total of 400 data points there are only 35 points that didn't have a higher than .5 chance of getting admitted.

```
# your code here
df$Research <- factor(df$Research)
df$Serial.No. <- NULL
df$Admit <- FALSE
df$Admit[df$Chance.of.Admit > .5] <- TRUE
df$Admit <- factor(df$Admit)
names(df)
```

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

```
# 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    Admit
##  Min.   :1.000   Min.   :6.800   0:181    Min.   :0.3400   FALSE: 35
##  1st Qu.:3.000   1st Qu.:8.170   1:219    1st Qu.:0.6400   TRUE :365
##  Median :3.500   Median :8.610            Median :0.7300
##  Mean   :3.453   Mean   :8.599            Mean   :0.7244
##  3rd Qu.:4.000   3rd Qu.:9.062            3rd Qu.:0.8300
##  Max.   :5.000   Max.   :9.920            Max.   :0.9700
```

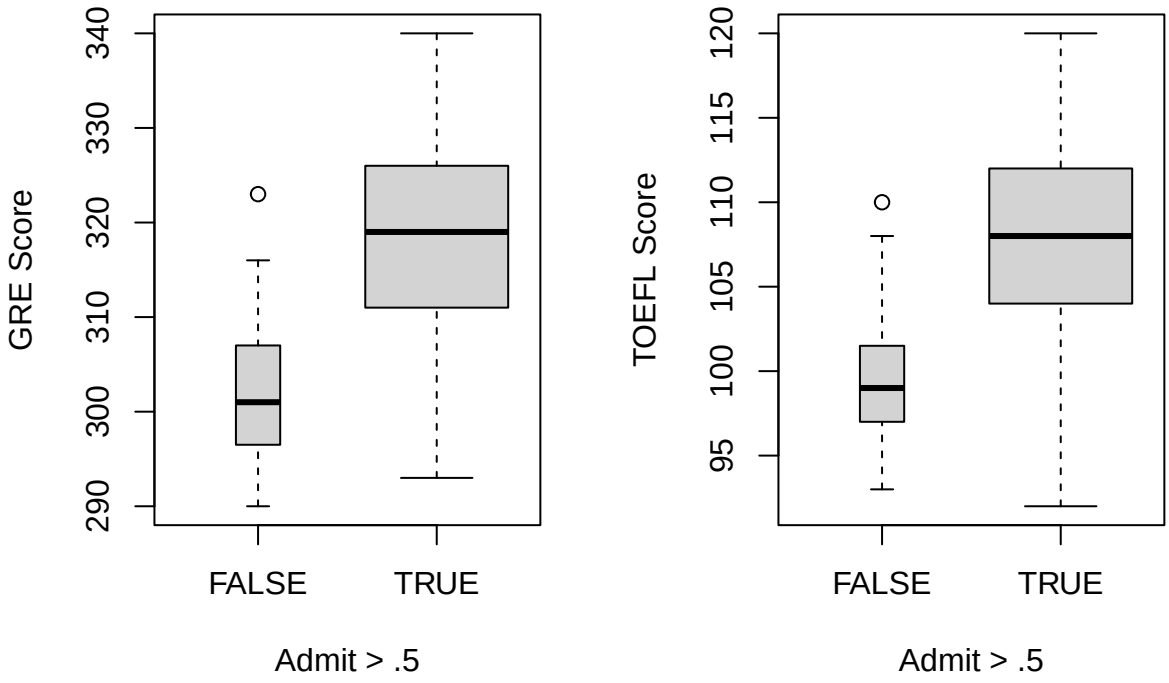
Step 3 Data Visualization

- 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:

```
# your code here
par(mfrow=c(1,2))
plot(df$GRE.Score~df$Admit, xlab = "Admit > .5", ylab = "GRE Score",varwidth=TRUE)
plot(df$TOEFL.Score~df$Admit, xlab = "Admit > .5", ylab = "TOEFL Score",varwidth=TRUE)
```



Step 4 Divide train/test

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

```
# your code here
set.seed(1234)
i <- sample(1:nrow(df)*0.75,replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

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: Yes I got 2 errors Warning: glm.fit: algorithm did not converge Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred. This is because the training data is too perfect or almost linearly perfect. This is because all the predictors were used.

```
# your code here

#glm1 <- glm(Admit~GRE.Score+TOEFL.Score+University.Rating+SOP+LOR+CGPA+Research+Chance.of.Admit, data=
glm1 <- glm(Admit~. ,data = train, family = "binomial")
```

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

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

```
summary(glm1)
```

```
##
## Call:
## glm(formula = Admit ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -1.916e-04   2.100e-08   2.100e-08   2.100e-08   2.292e-04
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -9.405e+02  2.553e+05  -0.004    0.997
## GRE.Score          8.271e-01  9.607e+02   0.001    0.999
## TOEFL.Score        8.595e-01  2.492e+03   0.000    1.000
## University.Rating -2.159e+01  9.435e+03  -0.002    0.998
## SOP               -2.953e+00  9.267e+03   0.000    1.000
## LOR                1.979e+01  8.243e+03   0.002    0.998
## CGPA              -1.531e+01  1.271e+04  -0.001    0.999
## Research1          1.248e+00  9.300e+03   0.000    1.000
## Chance.of.Admit    1.411e+03  1.588e+05   0.009    0.993
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.1793e+02  on 398  degrees of freedom
## Residual deviance: 2.0449e-07  on 390  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 was no error since we didn't include the Chance.of.Admit predictor.

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

```
##
## Call:
## glm(formula = Admit ~ . - Chance.of.Admit, family = binomial,
##     data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.06088   0.03489   0.09819   0.29025   1.34039
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -55.06576   11.10466  -4.959 7.09e-07 ***
## GRE.Score           0.06693    0.03976   1.683  0.09231 .
## TOEFL.Score         0.15557    0.08451   1.841  0.06565 .
## University.Rating  -0.60962    0.31843  -1.914  0.05556 .
## SOP                -0.54241    0.31722  -1.710  0.08729 .
## LOR                 1.25953    0.41664   3.023  0.00250 **
## CGPA                2.47326    0.78438   3.153  0.00162 **
## Research1          -0.45861    0.55721  -0.823  0.41048
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 217.93  on 398  degrees of freedom
## Residual deviance: 123.45  on 391  degrees of freedom
## AIC: 139.45
##
## Number of Fisher Scoring iterations: 7
```

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:

```r
# your code here
library(ROCR)
probs <- predict(glm1, newdata=test, type="response")
pr <- prediction(probs,test$Admit)
corr <- cor(probs,test$Chance.of.Admit)
print(corr)
```

```
## [1] 0.6979692
```

Step 8 Make binary predictions, print table and accuracy

- Run predict() again, this time making 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: Yes it was. It shows that the model is very accurate and that it only predicted incorrectly once where it predicted the Admit to be higher than .5 when it wasn't.

```
# your code here
pred <- ifelse(probs > .5, 2,1)
acc1 <- mean(pred==as.integer(test$Admit))
print(paste("glm1 accuracy = ",acc1))
```

```
## [1] "glm1 accuracy =  0.99"
```

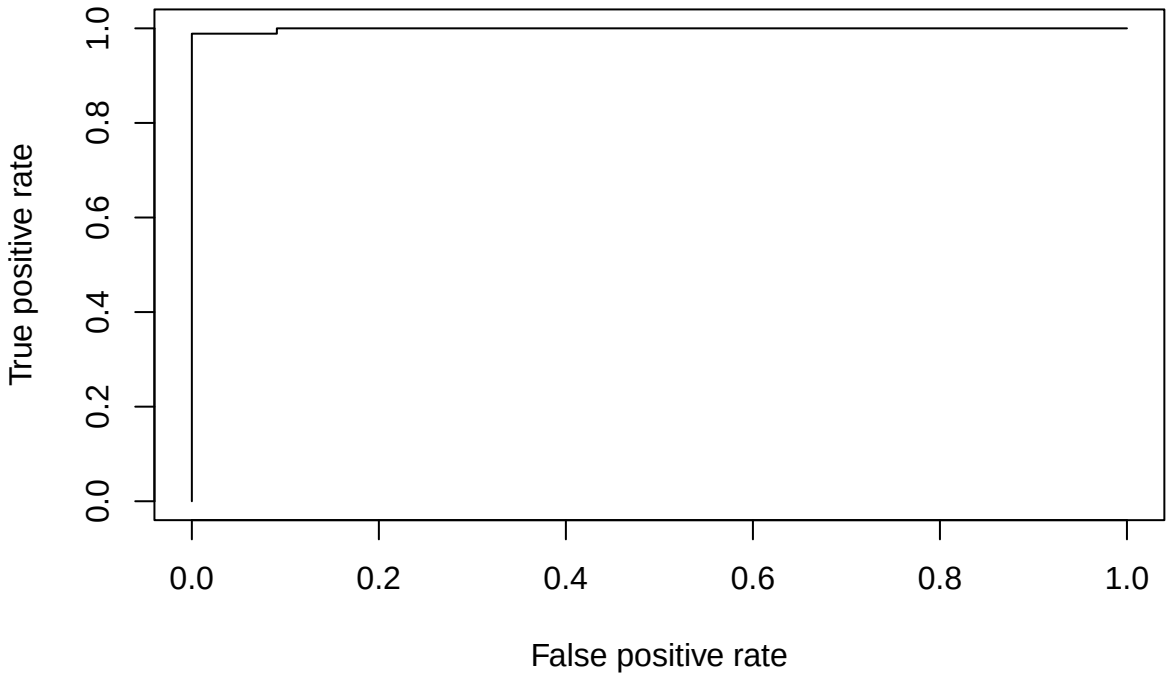
```
table(pred,as.integer(test$Admit))
```

```
##
## pred  1  2
##    1 11  1
##    2  0 88
```

Step 9 Output ROCR and AUC

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

```
# your code here
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
print(auc)
```

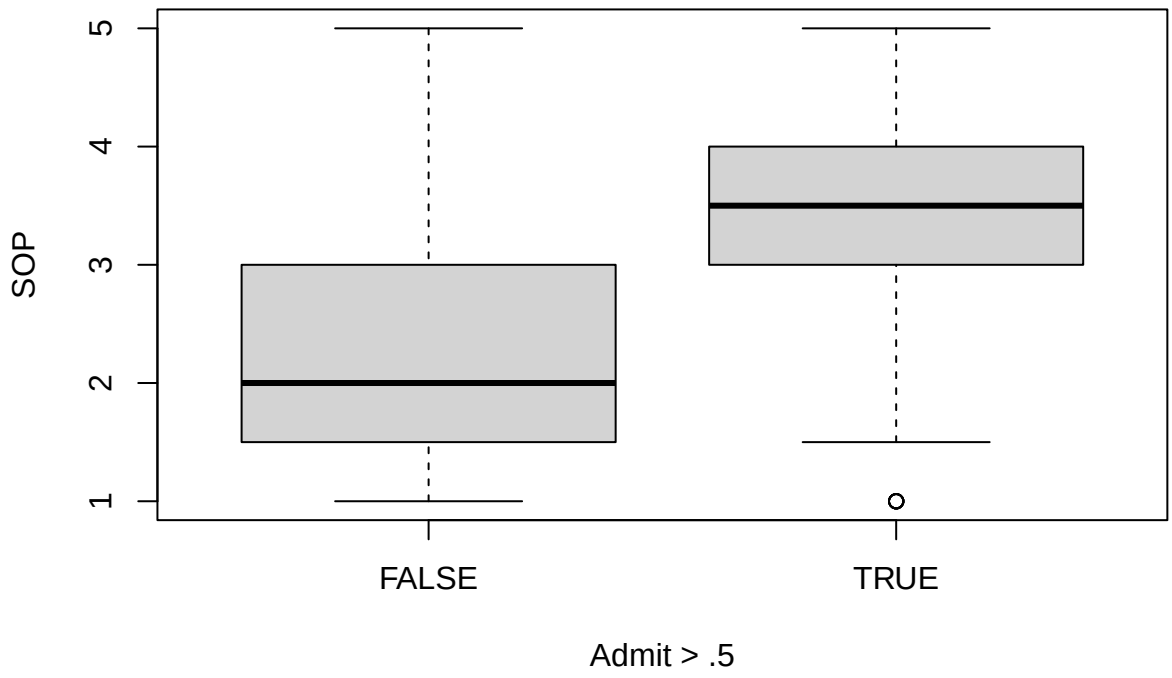
```
## [1] 0.9989785
```

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

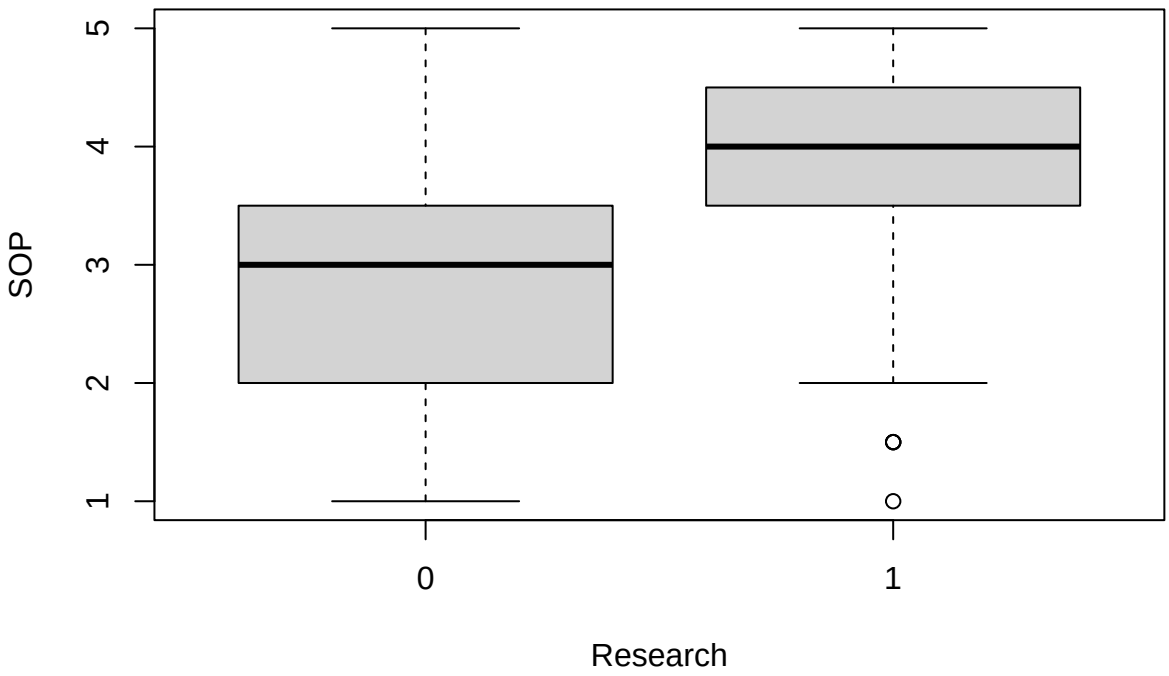
Your commentary here: For both having a higher chance at admission and having more research both correlate to having higher SOP.

```
# plot 1
plot(df$SOP~df$Admit,xlab = "Admit > .5", ylab = "SOP" )
```




```
# plot 2

plot(df$SOP~df$Research,xlab = "Research", ylab = "SOP" )
```


SOP
1
2
3
4
5
0
1
Research