Module 3 Assignment: GLM and Logistic Regression

William Scott Breckwoldt

College of Professional Studies: Analytics, Northeastern University

ALY 6015: Intermediate Analytics

Professor Eric Gero

January 30, 2022

Module 3 Assignment: GLM and Logistic Regression

Introduction

The GLM and Logistics Regression assignment includes the analysis and interpretations of a dataset that contains statistics from many US Colleges from 1995 courtesy of US News and World Report.

In this assignment, I will use R to process, analyze, and depict data, develop advanced models, using advanced generalized linear methods to answer strategic and operational questions, preparing complex datasets for analysis, and using multivariable and logistic regression method to improve predictive outcomes.

The main objective of this assignment is to build a logistic regression model to predict whether a university is private or public. I will interpret the results while using the steps outlined by Professor Gero in the lab and assignment. After the analysis, I will provide a conclusion in which will include my overall interpretations of the logistic regression model and the R code.

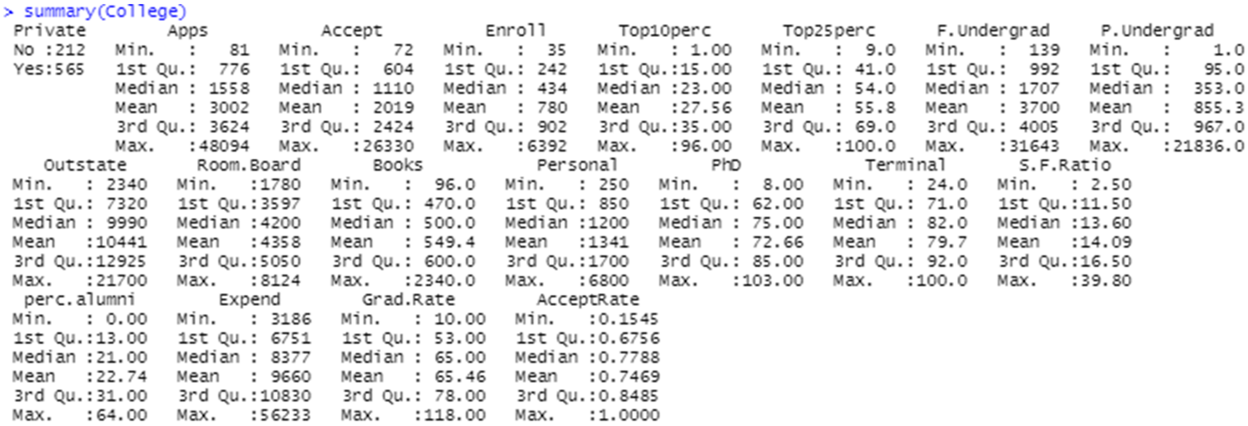
Analysis

1. Importing the Dataset and Performing Exploratory Analysis

I imported the dataset simply using attach(College). Then, to get an overview of the dataset, I used View(College). The environments tab told me that there are a total of 777 observations and 18 variables. Then, the descriptive statistics were calculated for the all variables using summary(College).

Figure 1

summary(College) R output



```
> summary(College)
 Private        Apps           Accept          Enroll       Top10perc       Top25perc      F.Undergrad     P.Undergrad     
 No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0  
 Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0  
           Median : 1558   Median : 1110   Median : 434   Median :23.00   Median : 54.0   Median : 1707   Median :  353.0  
           Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3  
           3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0  
           Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0  
    Outstate       Room.Board       Books           Personal         PhD            Terminal       S.F.Ratio    
 Min.   : 2340   Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50  
 1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50  
 Median : 9990   Median :4200   Median : 500.0   Median :1200   Median : 75.00   Median : 82.0   Median :13.60  
 Mean   :10441   Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09  
 3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50  
 Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80  
  perc.alumni        Expend        Grad.Rate        AcceptRate    
 Min.   : 0.00   Min.   : 3186   Min.   : 10.00   Min.   :0.1545  
 1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00   1st Qu.:0.6756  
 Median :21.00   Median : 8377   Median : 65.00   Median :0.7788  
 Mean   :22.74   Mean   : 9660   Mean   : 65.46   Mean   :0.7469  
 3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00   3rd Qu.:0.8485  
 Max.   :64.00   Max.   :56233   Max.   :118.00   Max.   :1.0000  
```

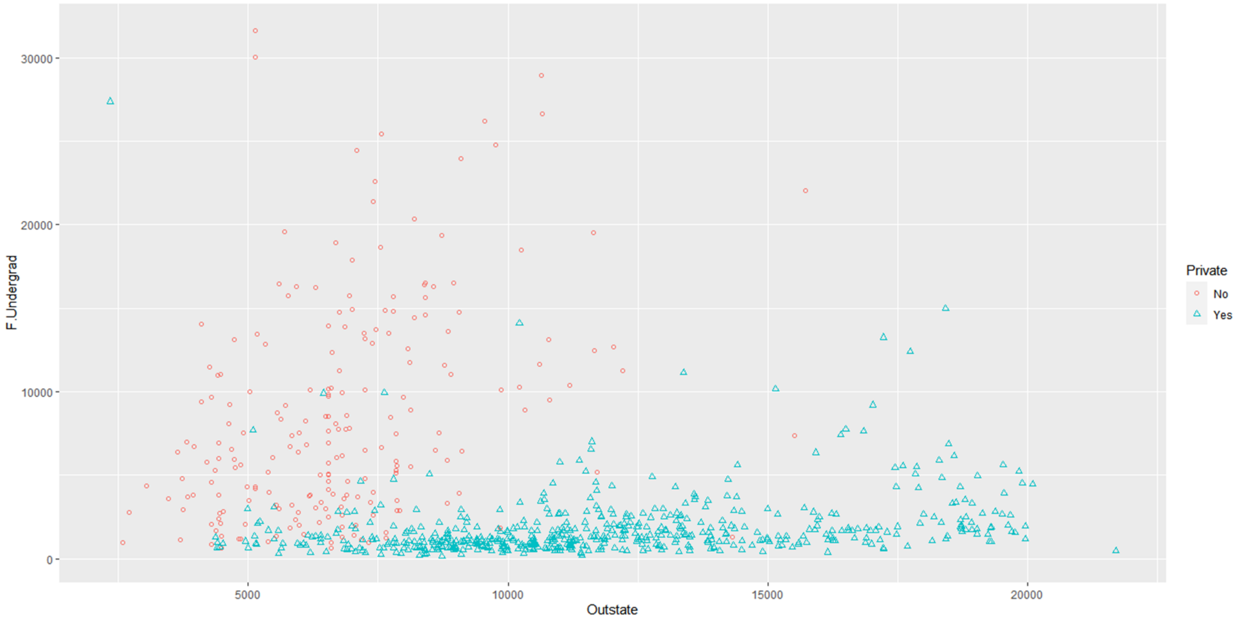
The variable I will focus most on is Private. This variable shows whether a university is private (Yes) or public (No). We can see that most of the universities from the dataset are private universities (565 compared to just 212).

I added a new variable called AcceptRate which was created using College$AcceptRate <- College$Accept/College$Apps. We can see that the lowest acceptance rate is 15.45%, the first quartile is 67.56%, the median is 77.88%, the mean is 74.69%, the third quartile is 84.85%, and the highest is 100%.

I also plotted some variables for review.

Figure 2

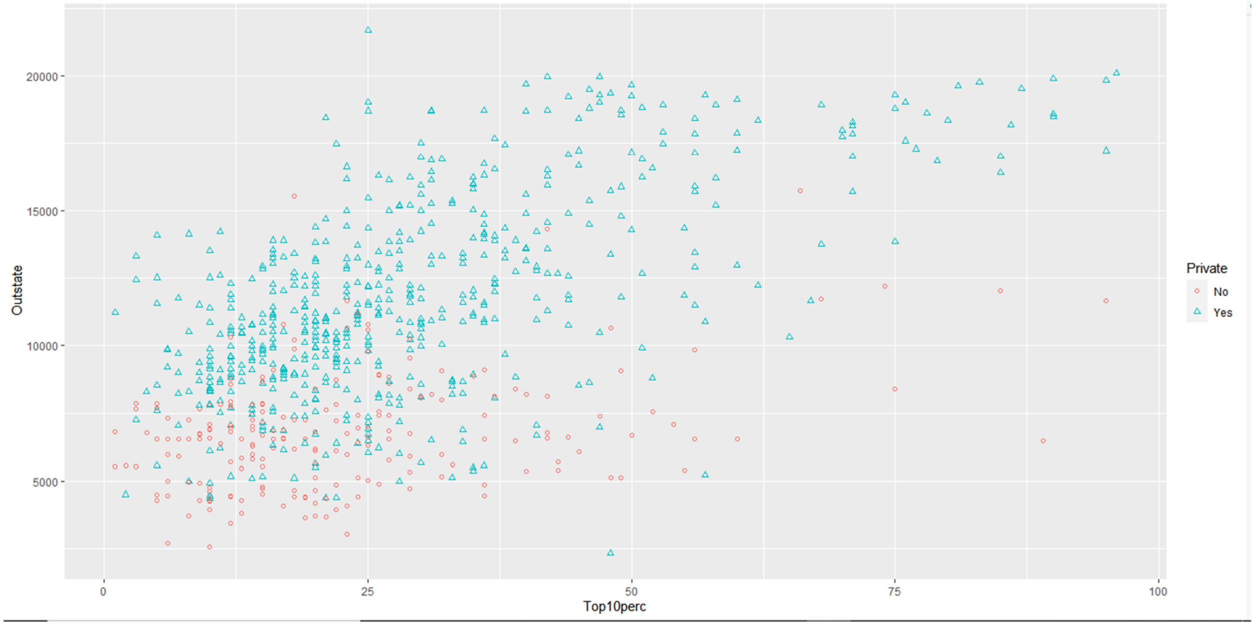
Full-Time Undergraduate Students vs. Out-of-State Scatter Plot



It appears with public universities tuition tends to rise with the number of full-time undergraduate students. On the other hand, it seems that private schools, that appear to be more expensive, have tuitions that are less affected by the number of full-time graduate students.

Figure 3

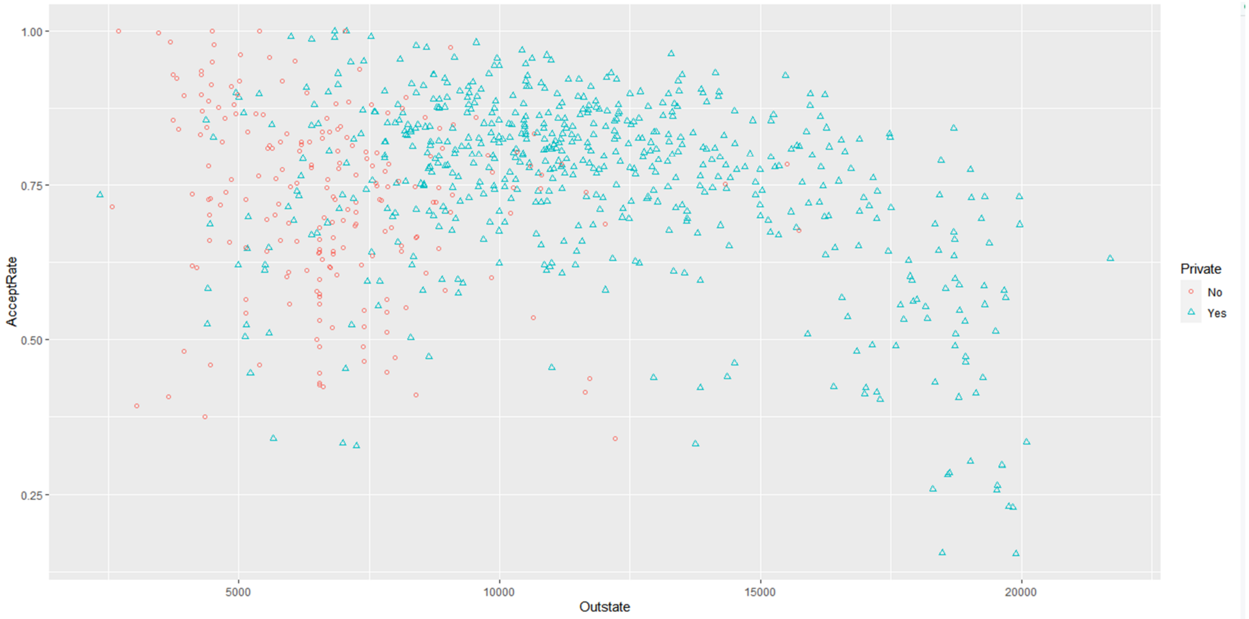
Out-of-State Tuition vs. Top 10 Percent Highschool Students Scatter Plot



It appears as the number of students who were in the top 10 percent of their class in high school rises, so does out-of-state tuition for both private and public universities.

Figure 4

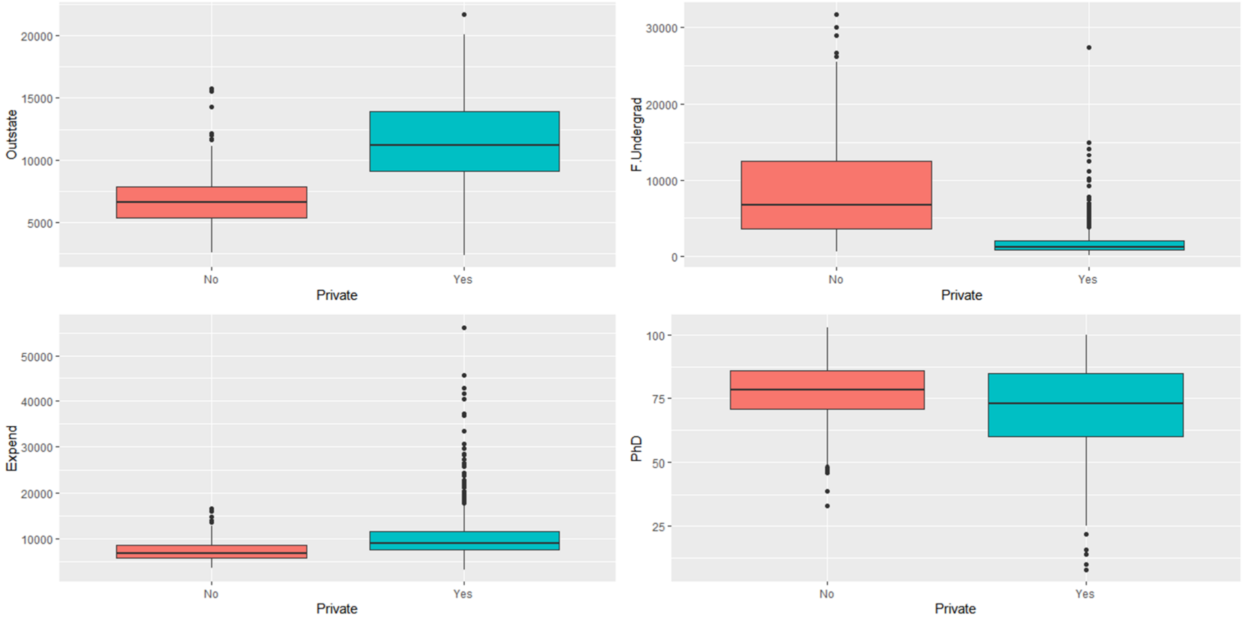
Acceptance Rate vs. Out-of-State Tuition



We can see that as the Out-of-State tuition increases, the acceptance rate generally declines.

Figure 6

Box Plots



Here are box plots showing the five number summaries. The out-of-state boxplot shows us that the minimums are similar, however, private schools generally have much higher out-of-state tuition. Additionally, we see that full-time undergraduate student populations are significantly larger at public schools. Expenditure per student is larger at private schools as well. Oddly, public universities have more professors with PhDs.

2. Splitting the Data in a Train and Test Set

Next, I will create the training and test sets. First, I will set the seed to randomize the results.

set.seed(123)

Then, I will pass the createDataPartition() function and include the response variable, the size of the partition I would like (70%) to be the training set, and I will call for a data frame, not a list.

trainIndex <- createDataPartition(College$Private, p = 0.70, list = FALSE)

Now that the training index is defined, I will pass that to College and assign it to the training set.

train <- College[trainIndex,]

I will also assign the indexes not within the train index to test.

test <- College[-trainIndex,]

3. Fit a Logistic Regression Model

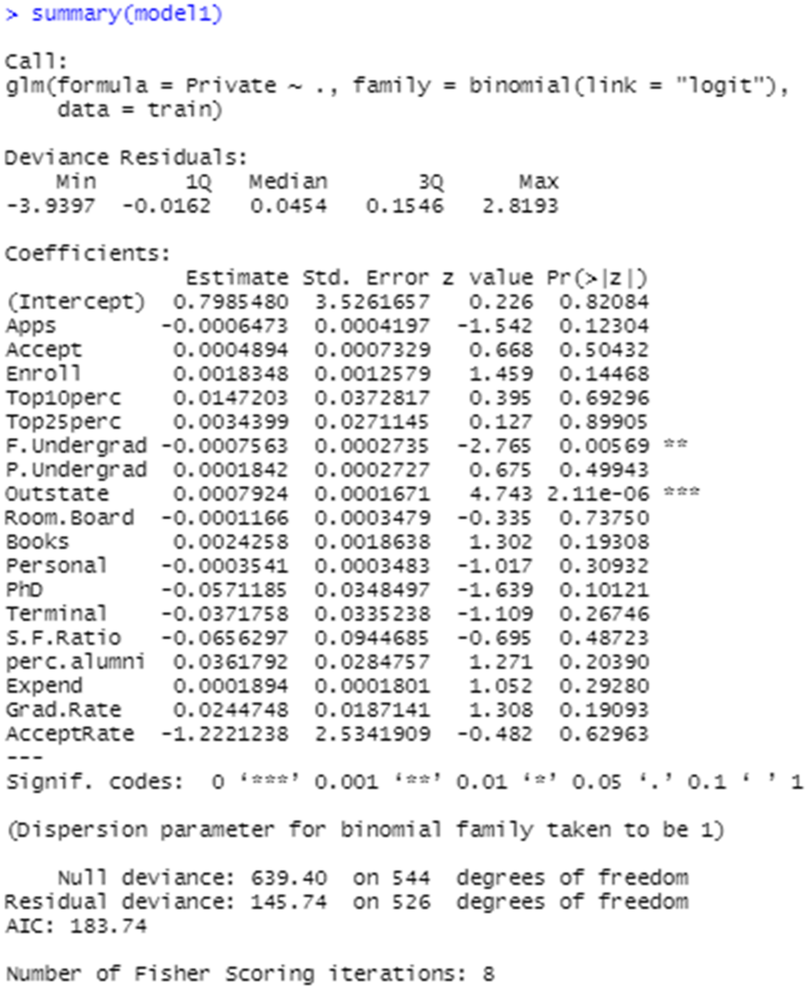
Now that I have defined my train and test set, I am ready to build my GLM models. First, I will run the regression model versus all predictors in the model signified by "." with my train dataset using the binomial family with the logic link.

model1 <- glm(Private ~ ., data = train, family = binomial(link = "logit"))

summary(model1)

Figure 7

Summary(model1) R Output



```
> summary(model1)

Call:
glm(formula = Private ~ ., family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9397  -0.0162   0.0454   0.1546   2.8193

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.7985480  3.5261657   0.226  0.82084
Apps        -0.0006473  0.0004197  -1.542  0.12304
Accept       0.0004894  0.0007329   0.668  0.50432
Enroll       0.0018348  0.0012579   1.459  0.14468
Top10perc    0.0147203  0.0372817   0.395  0.69296
Top25perc    0.0034399  0.0271145   0.127  0.89905
F.Undergrad -0.0007563  0.0002735  -2.765  0.00569 **
P.Undergrad  0.0001842  0.0002727   0.675  0.49943
Outstate     0.0007924  0.0001671   4.743 2.11e-06 ***
Room.Board  -0.0001166  0.0003479  -0.335  0.73750
Books        0.0024258  0.0018638   1.302  0.19308
Personal    -0.0003541  0.0003483  -1.017  0.30932
PhD         -0.0571185  0.0348497  -1.639  0.10121
Terminal    -0.0371758  0.0335238  -1.109  0.26746
S.F.Ratio   -0.0656297  0.0944685  -0.695  0.48723
perc.alumni  0.0361792  0.0284757   1.271  0.20390
Expend       0.0001894  0.0001801   1.052  0.29280
Grad.Rate    0.0244748  0.0187141   1.308  0.19093
AcceptRate  -1.2221238  2.5341909  -0.482  0.62963
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 639.40  on 544  degrees of freedom
Residual deviance: 145.74  on 526  degrees of freedom
AIC: 183.74

Number of Fisher Scoring iterations: 8
```

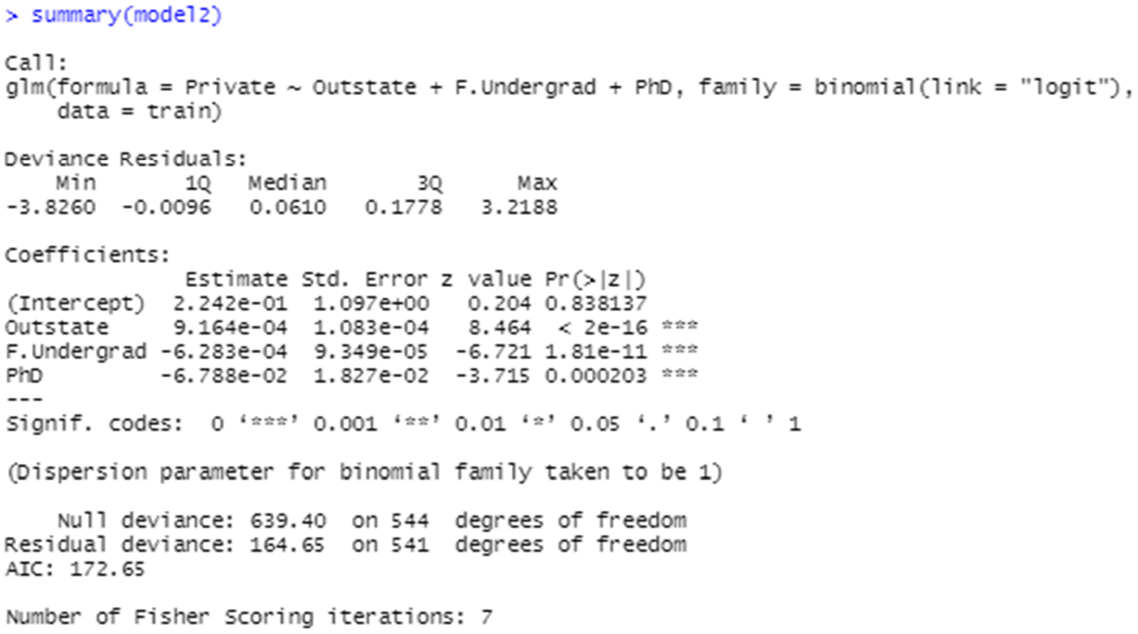
Looking at my output, we can see that F.Undergrad and Outstate have significant Pr(>|z|) values. The next model I fit will include those two predictors. I will also include PhD because it nearly missed the significant value of <0.1 and it also lowers the AIC when added(187.97 to 172.65).

model2<- glm(Private ~ Outstate + F.Undergrad + PhD, data = train, family = binomial(link = "logit"))

summary(model2)

Figure 8

Summary(model2) R Output



```
> summary(model2)

Call:
glm(formula = Private ~ Outstate + F.Undergrad + PhD, family = binomial(link = "logit"),
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8260  -0.0096   0.0610   0.1778   3.2188

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.242e-01  1.097e+00   0.204 0.838137
Outstate     9.164e-04  1.083e-04   8.464  < 2e-16 ***
F.Undergrad -6.283e-04  9.349e-05  -6.721 1.81e-11 ***
PhD         -6.788e-02  1.827e-02  -3.715 0.000203 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 639.40  on 544  degrees of freedom
Residual deviance: 164.65  on 541  degrees of freedom
AIC: 172.65

Number of Fisher Scoring iterations: 7
```

Despite having a lower AIC, the deviance difference is less than the model with all predictors, however, neither have a significant difference, suggesting that the models could be equally accurate.

4. Create a Confusion Matrix for the Train Set

First, I create my list of probabilities using the predict() function.

probabilities.train <- predict(model2, newdata = train, type = "response")

Using the ifelse() function and passing the probabilities list I just created, I am going to label anything with a probability of .5 and greater as a “Yes” and anything less than .5 as a No.

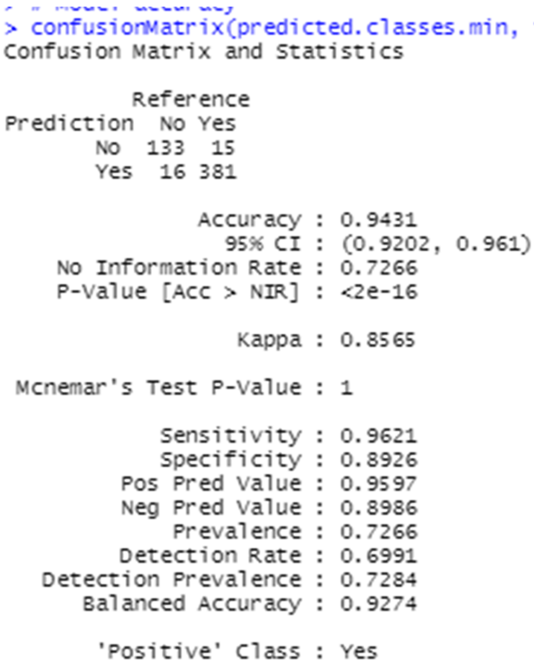
predicted.classes.min <- as.factor(ifelse(probabilities.train >= 0.5, "Yes", "No"))

Then, I will model for accuracy using the a confusion matrix.

confusionMatrix(predicted.classes.min, train$Private, positive = 'Yes')

Figure 9

confusionMatrix(predicted.classes.min, train$Private, positive = 'Yes') R Output



```
> # Model accuracy
> confusionMatrix(predicted.classes.min, train$Private,
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  133  15
       Yes  16 381

               Accuracy : 0.9431
                 95% CI : (0.9202, 0.961)
    No Information Rate : 0.7266
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8565

 Mcnemar's Test P-Value : 1

            Sensitivity : 0.9621
            Specificity : 0.8926
         Pos Pred Value : 0.9597
         Neg Pred Value : 0.8986
             Prevalence : 0.7266
         Detection Rate : 0.6991
   Detection Prevalence : 0.7284
      Balanced Accuracy : 0.9274

       'Positive' Class : Yes
```

According to the results, I have 133 true negatives, 381 true positives, 15 false negatives, and 16 false positives. For this dataset, there is no defined audience, so it is hard to say what misclassification is more damaging. I would say that false negatives and positives are similarly damaging. However, false positives seem to be more damaging for interested candidates looking to enroll in a university because there are fewer public schools in the dataset and public schools have larger populations and generally provide greater benefits to their instate students, like lower tuition, which is not accounted for in this model. Labeling a public school as private according to these predictors may lead candidates to lose interest who otherwise would have applied. Furthermore, the specificity of the model was lower than other metrics, which I will explain below.

5. Report Metrics for Accuracy, Precision, Recall, and Specificity

The results of the confusion matrix also provide metrics of accuracy, precision, recall, and specificity. Below, I will list the formulas and my calculations

TN = 133, TP = 381, FN = 15, FP = 16

Accuracy = (TN + TP)/(TN + FP + FN + TP) = (133 + 381) / (133 +169 + 15 + 381) = 0.9431

This implies that 94.31% of the predictions from my train set were correct.

Precision = TP/(FP + TP) = 381 / (16 + 381) = 0.9597

A precision of 0.9597 means that 95.97% of my positive predictions that a university was private were correct.

Recall = TP/(TP+FN) = 381 / (381 + 15) = 0.9621

A recall of 0.9621 means that of all the private schools, I predicted that they would be private 96.21% of the time.

Specificity = TN/(TN + FP) = 133 / (133 + 16) = 0.8926

A specificity of 0.8926 means that of all the public schools, I predicted they would be public 89.26% of the time.

6. Create a Confusion Matrix for the Test Set

Using the same method, I used in step 4, I will create a confusion matrix for the test set.

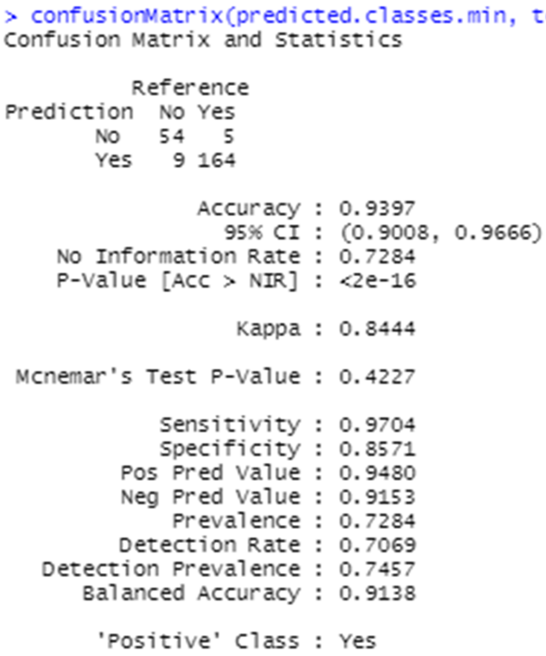
probabilities.test <- predict(model2, newdata = test, type = 'response')

predicted.classes.min <- as.factor(ifelse(probabilities.test >= 0.5, "Yes", "No"))

confusionMatrix(predicted.classes.min, test$Private, positive = 'Yes')

Figure 10

confusionMatrix(predicted.classes.min, test$Private, positive = 'Yes') R Output



```
> confusionMatrix(predicted.classes.min, test$Pr
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No   54   5
       Yes   9 164

               Accuracy : 0.9397
                 95% CI : (0.9008, 0.9666)
    No Information Rate : 0.7284
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8444

 Mcnemar's Test P-Value : 0.4227

            Sensitivity : 0.9704
            Specificity : 0.8571
         Pos Pred Value : 0.9480
         Neg Pred Value : 0.9153
             Prevalence : 0.7284
         Detection Rate : 0.7069
   Detection Prevalence : 0.7457
      Balanced Accuracy : 0.9138

       'Positive' Class : Yes
```

Using accuracy as the measurement metric, we can see that the two models are very similar suggesting that the model is not overfitted. Unfortunately, the specificity is somewhat low at 0.8571, which is not ideal but also not very worrisome.

7. Plot the ROC Curve

Using the roc() function from the pROC library, I pass in the test$Private, my response variable, as well as the predicted probabilities.

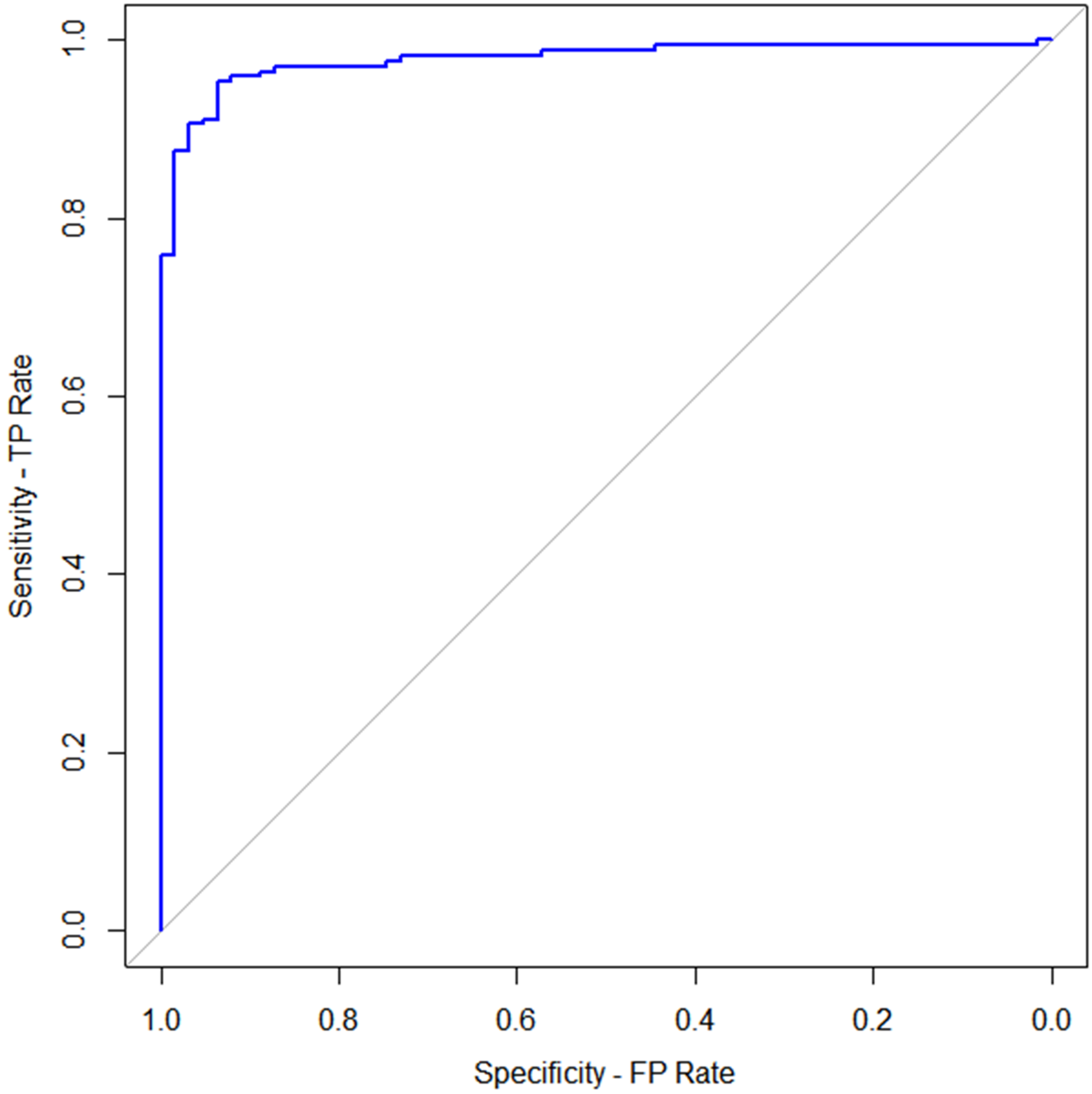
ROC1 <- roc(test$Private, probabilities.test)

Then, I plot my ROC1 object.

plot(ROC1, col = "blue", ylab = "Sensitivity - TP Rate", xlab = 'Specificity - FP Rate')

Figure 11

The Receiver Operator Characteristic Curve



The plot shows sensitivity versus specificity. Since the blue line is far from the diagonal and hugs the borders, it suggests that the model is accurate. We can also tell from the plot that the sensitivity or TP rate had a better outcome than the specificity or FP rate.

8. **Calculate the AUC**

I calculate the area under the receiver operator characteristic curve using the auc() function.

```
auc <- auc(ROC1)

auc

Area under the curve: 0.9776
```

This measurement once again suggests that the model used in this assignment was quite accurate.

Conclusion

Following the steps provided by Professor Gero, I arrived upon a logistics regression model that performs classification for private and public universities using predictors that I am confident have had an accurate outcome. I also completed my goals of using R to process, analyze and depict data, develop advanced models, use advanced generalized linear methods, prepare complex data for analysis, and use multivariable and logistic regression method to improve predictive outcomes.

One note: as I was unsure whom the audience is (maybe people interested in pursuing a university degree or university representatives), it was hard to determine what misclassification would be more damaging for the analysis in step 4. However, the overall accuracy of the model was impressive, so the damage may not be significant, except for specificity, as the confusion matrix for the test set suggested that of all the public schools, I predicted they were public schools just 85.71% of the time.

It would be interesting to see how this model could be applied to a real-world scenario. For instance, in the final group project, could this method be applied to a housing dataset to predict housing categories? Or maybe in a sports dataset to predict left-handed batters or pitchers? This will be worth considering as the final project deadline approaches.

Appendix

```
library(ISLR)

library(caret)

library(ggplot2)

library(gridExtra)

library(pROC)

library(stats)

#1 Import the dataset and perform Exploratory Data Analysis by using descriptive statistics and plots to describe the dataset.

####################################################

# Load Data
```

```
##################################################
attach(College)


head(College)


summary(College)


College$AcceptRate <- College$Accept/College$Apps


View(College)


##################################################
# Plot some variables for review
##################################################
# Scatter plot
qplot(x = Top10perc, y = Outstate, color = Private, shape = Private, geom = 'point') +
scale_shape(solid = FALSE)


qplot(x = Grad.Rate, y = Outstate, color = Private, shape = Private, geom = 'point') +
scale_shape(solid = FALSE)


qplot(x = Outstate, y = Expend, color = Private, shape = Private, geom = 'point') +
scale_shape(solid = FALSE)


qplot(x = Outstate, y = F.Undergrad, color = Private, shape = Private, geom = 'point') +
scale_shape(solid = FALSE)


qplot(x = Outstate, y = AcceptRate, color = Private, shape = Private, geom = 'point') +
scale_shape(solid = FALSE)
```

```
# Box plots
w <- qplot(x = Private, y = Outstate, fill = Private, geom = 'boxplot') + guides(fill = "none")
x <- qplot(x = Private, y = Expend, fill = Private, geom = 'boxplot') + guides(fill = "none")
y <- qplot(x = Private, y = PhD, fill = Private, geom = 'boxplot') + guides(fill = "none")
z <- qplot(x = Private, y = F.Undergrad, fill = Private, geom = 'boxplot') + guides(fill = "none")
grid.arrange(w, z, x, y, nrow = 2)
##################################################
#2 Split data into train and test sets
##################################################
set.seed(123)
trainIndex <- createDataPartition(College$Private, p = 0.70, list = FALSE)
train <- College[trainIndex,]
test <- College[-trainIndex,]

head(train)

##################################################
#3 Use the glm() function in the 'stats' package to fit a logistic regression model to the training set using at least two predictors.
##################################################
model1 <- glm(Private ~ ., data = train, family = binomial(link = "logit"))
summary(model1)

model2<- glm(Private ~ Outstate + F.Undergrad + PhD, data = train, family = binomial(link = "logit"))
summary(model2)

#Display the regression coefficients (log-odds)
coef(model2)
```

```
# Display regression coefficients (odds)

exp(coef(model2))


# Odds of private school increase by a factor of 1.00079824 for each 1 unit increase in Outstate

# Odds of default increase by a factor of 0.99925923 for each 1 unit increase in full-time undergrads
```

#4 Create a confusion matrix and report the results of your model for the train set. Interpret and discuss the confusion matrix. Which misclassifications are more damaging for the analysis, False Positives or False Negatives?

```
####################################################

# Train set predictions

####################################################

# Make predictions on the test data using lamba.min

probabilities.train <- predict(model2, newdata = train, type = "response")

predicted.classes.min <- as.factor(ifelse(probabilities.train >= 0.5, "Yes", "No"))


# Model accuracy

confusionMatrix(predicted.classes.min, train$Private, positive = 'Yes')


# False negative (FN) = predicted no, actually is yes

# False positive (FP) = predicted yes, actually is no


# Would say FP are more dangerous because there are much less public schools in the dataset (212/(212+565)) instate vs outstate tuition
```

#5 Report and interpret metrics for Accuracy, Precision, Recall, and Specificity.

```
# Accuracy = (TN + TP)/(TN + FP + FN + TP)
# Precision = TP/(FP + TP)
# Recall = TP/(TP+FN)
# Sensitivity = TP rate
# Specificity = TN/(TN + FP)

#6 Create a confusion matrix and report the results of your model for the test set.
########################################################
# Test set predictions
########################################################
probabilities.test <- predict(model2, newdata = test, type = 'response')
predicted.classes.min <- as.factor(ifelse(probabilities.test >= 0.5, "Yes", "No"))

# Model accuracy
confusionMatrix(predicted.classes.min, test$Private, positive = 'Yes')

#7 Plot and interpret the ROC curve.
# Plot the Receiver operator characteristic curve
ROC1 <- roc(test$Private, probabilities.test)

plot(ROC1, col = "blue", ylab = "Sensitivity - TP Rate", xlab = 'Specificity - FP Rate')

#8 Calculate and interpret the AUC.
#Calculate the area under the ROC curve
auc <- auc(ROC1)
auc
```