Homework 9

Statistical Computing, STAT 3675Q

Anthony Anderson

General Instructions

- Answer the questions by inserting R code and necessary comments. Your output must contain the R code (do not use the `echo=FALSE option`).
- After you complete the assignment, save it under the file name `LastName-FirstName-HW9.pdf`
- Then submit the compiled PDF file through HuskyCT by **May 2, 2019, at 11:59 PM**.

Data

This problem relates to the College data set in the **ISLR** package. It contains a number of variables for 777 different universities and colleges in the US. The variables are

- **Private** : Public/private indicator
- **Apps** : Number of applications received
- **Accept** : Number of applicants accepted
- **Enroll** : Number of new students enrolled
- **Top10perc** : New students from top 10% of high school class
- **Top25perc** : New students from top 25% of high school class
- **F.Undergrad** : Number of full-time undergraduates
- **P.Undergrad** : Number of part-time undergraduates
- **Outstate** : Out-of-state tuition
- **Room.Board** : Room and board costs
- **Books** : Estimated book costs
- **Personal** : Estimated personal spending
- **PhD** : Percent of faculty with Ph.D.'s
- **Terminal** : Percent of faculty with terminal degree
- **S.F.Ratio** : Student/faculty ratio
- **perc.alumni** : Percent of alumni who donate
- **Expend** : Instructional expenditure per student
- **Grad.Rate** : Graduation rate

In this problem, you will predict if an institution is private or public using the other variables.

a. [**2**] Split the data into a training set and a validation set. The training set contains 80% of the data points, and the validation set contains the remaining observations.

```
library(ISLR)
college<-College
set.seed(42)
train=sample(777,.8*nrow(college))
train.data <- College[train,]
test.data <- College[-train,]
```

b. [**2**] Perform a logistic regression using the training set.

```
log.fit<- glm(Private~.-Private,data=train.data, family = binomial())
summary(log.fit)
```

```
##
## Call:
## glm(formula = Private ~ . - Private, family = binomial(), data = train.data)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5772  -0.0122   0.0423   0.1603   2.8352
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.0315956  2.1166683   0.015  0.98809
## Apps        -0.0005999  0.0003000  -2.000  0.04552 *
## Accept       0.0009190  0.0005882   1.562  0.11821
## Enroll       0.0006469  0.0010320   0.627  0.53078
## Top10perc   -0.0127589  0.0333419  -0.383  0.70197
## Top25perc    0.0032354  0.0225421   0.144  0.88587
## F.Undergrad -0.0007471  0.0002398  -3.115  0.00184 **
## P.Undergrad  0.0001794  0.0001550   1.158  0.24704
## Outstate     0.0007426  0.0001362   5.453 4.95e-08 ***
## Room.Board  -0.0002215  0.0003300  -0.671  0.50217
## Books        0.0026807  0.0017786   1.507  0.13175
## Personal    -0.0002754  0.0003371  -0.817  0.41392
## PhD         -0.0782153  0.0307533  -2.543  0.01098 *
## Terminal    -0.0044206  0.0286427  -0.154  0.87734
## S.F.Ratio   -0.0926733  0.0658598  -1.407  0.15939
## perc.alumni  0.0537475  0.0267627   2.008  0.04461 *
## Expend       0.0001883  0.0001306   1.443  0.14912
## Grad.Rate    0.0240068  0.0143789   1.670  0.09500 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 729  on 620  degrees of freedom
## Residual deviance: 174  on 603  degrees of freedom
## AIC: 210
##
## Number of Fisher Scoring iterations: 8
```

c. [**2**] If for a certain institution, the predicted probability is greater than 0.5, classify it as *private*. Using the model in b., predict the type of institution (Private/Public) for each observation in the validation set.

```
prob <-predict(log.fit,test.data,type="response")
log.pred<-factor(prob>.5, levels = c(FALSE,TRUE),labels=c("No","Yes"))
```

d. [**2**] Summarize the classification in a table, showing the true classification vs the classification obtained from the logistic regression model for the validation set.

```
log.error<- table(test.data$Private,log.pred,dnn=c("Actual","predicted"))
log.error
```

```
##       predicted
## Actual  No Yes
##    No   34   8
##    Yes   7 107
```

e [**6**]. Repeat b.,c.,d. using a decision tree.

```
library(rpart)
d.tree <- rpart(Private~.-Private,data=train.data,method="class", parms=list(split="information"))
```

```
d.tree.prob<-predict(d.tree,test.data,type="class")
d.tree.pred<-table(test.data$Private,d.tree.prob,dnn=c("Actual","predicted"))
d.tree.pred
```

```
##       predicted
## Actual  No Yes
##    No   32  10
##    Yes   6 108
```

f [**6**]. Repeat b.,c.,d. using a conditional inference tree.

```
library(party)
```

```
## Loading required package: grid
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: sandwich
```

```
c.tree<-ctree(Private~.-Private,data=train.data)
c.tree.prob<-predict(c.tree,test.data,type="response")
c.tree.pred<-table(test.data$Private,c.tree.prob,dnn=c("Actual","predicted"))
c.tree.pred
```

```
##       predicted
## Actual  No Yes
##    No   42   0
##    Yes   0 114
```

g [**6**]. Repeat b.,c.,d. using a random forest.

```
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
```

```
set.seed(42)
forest.fit<-randomForest(Private~.,data=train.data, na.action=na.roughfix ,importance=TRUE)
forest.prob<-predict(forest.fit,test.data)
forest.pred<-table(test.data$Private,forest.prob,dnn=c("Actual","predicted"))
forest.pred
```

```
##       predicted
## Actual  No Yes
##    No   36   6
##    Yes   3 111
```

h. [4] Compute sensitivity, specificity, positive predictive value, negative predictive value, and accuracy of each method using the function `performance()` shown on slide 45 in the lecture notes. Overall, which approach is the best?

```
performance <- function(table, n=2){
if(!all(dim(table) == c(2,2)))
stop("Must be a 2 x 2 table")
tn = table[1,1]
fp = table[1,2]
fn = table[2,1]
tp = table[2,2]
sensitivity = tp/(tp+fn)
specificity = tn/(tn+fp)
ppp = tp/(tp+fp)
npp = tn/(tn+fn)
hitrate = (tp+tn)/(tp+tn+fp+fn)
result <- paste("Sensitivity = ", round(sensitivity, n) ,
"\nSpecificity = ", round(specificity, n),
"\nPositive Predictive Value = ", round(ppp, n),
"\nNegative Predictive Value = ", round(npp, n),
"\nAccuracy = ", round(hitrate, n), "\n", sep="")
cat(result)
}
performance(log.error)
```

```
## Sensitivity = 0.94
## Specificity = 0.81
## Positive Predictive Value = 0.93
## Negative Predictive Value = 0.83
## Accuracy = 0.9
```

```
performance(d.tree.pred)
```

```
## Sensitivity = 0.95
## Specificity = 0.76
## Positive Predictive Value = 0.92
## Negative Predictive Value = 0.84
## Accuracy = 0.9
```

```
performance(c.tree.pred)
```

```
## Sensitivity = 1
## Specificity = 1
## Positive Predictive Value = 1
## Negative Predictive Value = 1
## Accuracy = 1
```

```
performance(forest.pred)
```

```
## Sensitivity = 0.97
## Specificity = 0.86
## Positive Predictive Value = 0.95
## Negative Predictive Value = 0.92
## Accuracy = 0.94
```

In order we have the performance of the logistic regression, decision tree, conditional inference tree, and the random forest.

The best performing method for this dataset is the conditional inference tree, with values of 1 for all performance statistics. It perfectly classified all true values as true, all false as false, with no errors. All true values are actually true, and the same goes for the false case. The next runner up would be the logistic regression, since overall each performance statistic is still very high. The forest suffers a (relative to the other methods) low specificity rating, it may misclassify more often than the logistic classifier.