DSI-06_Homework 5: Chapter 6, pg 286

Julia Gallucci

2023-03-05

13. In this exercise, we will predict the number of applications received using the other variables in the College data set.

```
#install.packages("ISLR") install package containing College dataset
library(ISLR) #load library
attach(College) #attach College dataset to make the variables associated with College available.
head(College) #return the column names and first few rows of the dataset
```

```
##                              Private Apps Accept Enroll Top10perc Top25perc
## Abilene Christian University     Yes 1660   1232    721        23        52
## Adelphi University               Yes 2186   1924    512        16        29
## Adrian College                   Yes 1428   1097    336        22        50
## Agnes Scott College              Yes  417    349    137        60        89
## Alaska Pacific University        Yes  193    146     55        16        44
## Albertson College                Yes  587    479    158        38        62
##                              F.Undergrad P.Undergrad Outstate Room.Board Books
## Abilene Christian University        2885         537     7440       3300   450
## Adelphi University                  2683        1227    12280       6450   750
## Adrian College                      1036          99    11250       3750   400
## Agnes Scott College                  510          63    12960       5450   450
## Alaska Pacific University            249         869     7560       4120   800
## Albertson College                    678          41    13500       3335   500
##                              Personal PhD Terminal S.F.Ratio perc.alumni Expend
## Abilene Christian University     2200  70       78      18.1          12   7041
## Adelphi University               1500  29       30      12.2          16  10527
## Adrian College                   1165  53       66      12.9          30   8735
## Agnes Scott College               875  92       97       7.7          37  19016
## Alaska Pacific University        1500  76       72      11.9           2  10922
## Albertson College                 675  67       73       9.4          11   9727
##                              Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
## Alaska Pacific University           15
## Albertson College                   55
```

(a) Split the data set into a training set and a test set.

We can use the function sample() to split the observations into two halves for the training and testing sets. The train vector contains the indices of the College dataset that will be used as the training data.

```r
# split the 777 observations in half (roughly) so 388 observations in the training set
train <- sample (1: nrow(College), nrow(College) / 2)
test <- (-train)
```

(b) Fit a linear model using least squares on the training set, and report the test error obtained.

```r
lm.fit <- lm(Apps ~ ., data = College, subset = train) # fit model on only training data
lm.pred <- predict(lm.fit, College[test,], type="response") #predict using our fit model on test data
mean((lm.pred - College[test,]$Apps )^2) #given our predictions we can compute the MSE for the test set
```

```
## [1] 1135758
```

So our estimated test MSE for the linear regression is 1135758

(c) Fit a ridge regression model on the training set, with lambda chosen by cross-validation. Report the test error obtained.

We will use the glmnet() function from the glmnet package to perform ridge regression. Instead of using a data frame, the glmnet() function requires that the predictors be in the form of a matrix and the response be in the form of a vector. The model.matrix() automatically transforms qualitative variables into dummy variables.

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-6
```

```r
#Set up matrices needed for the glmnet functions
train.mat = model.matrix(Apps~., data = College[train,]) #training matrix using train observations
test.mat = model.matrix(Apps~., data = College[test,]) #testing matrix using test observations
```

Choose lambda using cross-validation

```r
lamda_cv = cv.glmnet(train.mat,College[train,]$Apps,alpha=0) #perform cross validation on our model
bestlam = lamda_cv$lambda.min #minimum lambda extracted
bestlam
```

```
## [1] 405.8404
```

Fit a ridge regression model and predict on our testing data using our best lambda!

```r
ridge.mod = glmnet(train.mat,College[train,]$Apps,alpha = 0)
#Make predictions
ridge.pred = predict(ridge.mod,s=bestlam,newx = test.mat)

mean((ridge.pred - College[test,]$Apps)^2) #given our predictions we can compute the MSE for the test s
```

```
## [1] 976261.5
```

The test error of the ridge regression fit with a lambda chosen by cross-validation is 976261.5, lower than the linear model test error!

(d) Fit a lasso model on the training set, with lambda chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimates.

We will perform the lasso the exact same way as ridge regression except we have alpha = 1 in the glmnet() function to specify the lasso method is to be used.

Choose lambda using cross-validation

```
lamda_cv2 = cv.glmnet(train.mat,College[train,]$Apps,alpha=1)
bestlam_2 = lamda_cv2$lambda.min
bestlam_2
```

```
## [1] 1.97344
```

Fit a lasso regression model and predict on our testing data using our best lambda!

```
#Fit lasso model
lasso.mod = glmnet(train.mat,College[train,]$Apps,alpha = 1)
#Make predictions
lasso.pred = predict(lasso.mod,s=bestlam_2,newx=test.mat)
mean((lasso.pred - College[test,]$Apps)^2)
```

```
## [1] 1115901
```

The test error of the lasso model fit with a lambda chosen by cross-validation is 1115901 ,higher thanridge regression test error but lower than the linear model!

Based on test error, the model performance from best to worst is:

1. Ridge Regression (976261.5)
2. Lasso model (1115901)
3. Linear model (1135758)