PRACTICE HW2

Hillary

1/30/2020

(a) Split the data set into a training set and a test set using caret library and fit each of the following models using caret and ten fold cross validation.

```
library(ISLR)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 3.0-2

attach(College)
head(College)

##                              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
```

```
x <- model.matrix(Apps~., College)[,-1]
y <- College$Apps
lambda <- 10^seq(10, -2, length = 100)


# Train test split
set.seed(489)
train = sample(1:nrow(x), nrow(x)/2)
test = (-train)
ytest = y[test]
```

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

```
OLS_lm <- lm(Apps~., data = College, subset = train)
OLS_lm

##
## Call:
## lm(formula = Apps ~ ., data = College, subset = train)
##
## Coefficients:
## (Intercept)   PrivateYes       Accept       Enroll    Top10perc
Top25perc
##  -544.41744   -170.52279      1.74160     -1.41087     38.28257     -
6.06587
## F.Undergrad  P.Undergrad     Outstate   Room.Board        Books
Personal
##     0.07306      0.08748     -0.08632      0.16650      0.06319
0.09351
##         PhD     Terminal    S.F.Ratio  perc.alumni       Expend
```

```
Grad.Rate
##   -11.10782      2.19668      4.12585      3.56206      0.05095
1.92934

#Find the best lambda from our list via cross-validation
cv.out <- cv.glmnet(x[train,], y[train], alpha = 0)
cv.out

##
## Call:  cv.glmnet(x = x[train, ], y = y[train], alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Measure      SE Nonzero
## min  397.4 2103455 1270039      17
## 1se 2554.6 3360297 2169940      17

#Best lambda
bestlam <- cv.out$lambda.min
bestlam

## [1] 397.4201

#Make predictions
OLS.pred <- predict(OLS_lm, newdata = College[test,])
head(OLS.pred)

##        Adelphi University            Adrian College         Albertson
College
##                3350.61158                1397.93516
608.67123
##   Albertus Magnus College Alderson-Broaddus College         Allegheny
College
##                  54.98646                 686.22811
2922.74735

#check Mean Squared Error
mean((OLS.pred-ytest)^2)

## [1] 1403054
```

(c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test mean squared error obtained. Report the value of λ used in the model

```
ridge.mod <- glmnet(x[train,], y[train], alpha = 0, lambda = lambda)
summary(ridge.mod)

##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1700   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
```

```
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```

```
#Find the best lambda from our list via cross-validation
cv.out <- cv.glmnet(x[train,], y[train], alpha = 0)
cv.out
```

```
##
## Call:  cv.glmnet(x = x[train, ], y = y[train], alpha = 0)
##
## Measure: Mean-Squared Error
##
##     Lambda Measure      SE Nonzero
## min  397.4 2352967 1646036      17
## 1se 3077.1 3903384 2937840      17
```

```
#Best lambda
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 397.4201
```

```
#make predictions
ridge.pred <- predict(ridge.mod, s = bestlam, newx = x[test,])
head(ridge.pred)
```

```
##                                1
## Adelphi University        3000.9738
## Adrian College            1164.0138
## Albertson College          595.0114
## Albertus Magnus College    317.8752
## Alderson-Broaddus College  549.4096
## Allegheny College         2677.7668
```

```
#Mean squared error
mean((ridge.pred-ytest)^2)
```

```
## [1] 1298095
```

(d) Fit a lasso model on the training set, with fraction chosen by cross validation. Report the test mean squared error obtained, along with the number of non-zero coefficient estimates and the fraction.

```
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = lambda)
summary(lasso.mod)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
```

```
## beta      1700   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         5   -none-    call
## nobs         1   -none-    numeric
```

```
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[test,])
head(lasso.pred)
```

```
##                                  1
## Adelphi University        2741.3266
## Adrian College            1686.0656
## Albertson College          998.9299
## Albertus Magnus College    629.1303
## Alderson-Broaddus College  875.6115
## Allegheny College         2954.1927
```

```
mean((lasso.pred-ytest)^2)
```

```
## [1] 1798354
```

(e) Fit a PCR model on the training set, with no. of principal components M chosen by cross-validation. Report the test mean squared error obtained, along with the value of M selected by cross-validation.

```
set.seed(123)
smp_size <- floor(0.75 * nrow(mtcars))
train_ind <- sample(seq_len(nrow(College)), size = smp_size)
train_p <- College[train_ind, ]
test_p <- College[-train_ind,c(1,3:18) ]
y_test=College[-train_ind,2]

require(pls)
```

```
## Loading required package: pls
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
pcr_model <- pcr(Apps~., data = train_p,scale =TRUE, validation = "CV")
summary(pcr_model)
```

```
## Data:    X dimension: 24 17
##  Y dimension: 24 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            2426     2779     1375     1389     1371     1509     1612
## adjCV         2426     2749     1351     1365     1342     1477     1568
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1605     1625     1842      1786      1664      1373      1346
## adjCV     1559     1574     1779      1714      1606      1305      1282
##        14 comps  15 comps  16 comps  17 comps
## CV         1180     936.9      1293      2503
## adjCV      1126     890.5      1224      2386
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8
comps
## X       37.40    62.89    73.63    81.16    87.87    91.82    93.96
95.65
## Apps    23.12    81.60    82.53    84.39    86.72    89.36    90.93
91.92
##       9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X       97.14     97.93     98.65     99.09     99.50     99.79     99.96
## Apps    92.87     95.00     95.49     98.20     98.27     98.44     98.99
##       16 comps  17 comps
## X        99.98    100.00
## Apps     98.99     99.03

pcr_pred <- predict(pcr_model, test_p, ncomp = 3)
head(pcr_pred)

## [1] 1930.6961 1451.1950  704.8014 2322.1893  815.2842 1231.9749

mean((pcr_pred - y_test)^2)

## [1] 3664827
```

(f) Fit a PLS model on the training set, with M chosen by cross validation. Report the test error obtained, along with the value of M selected by cross-validation.

```
library(caret)

## Loading required package: lattice

## Loading required package: ggplot2

##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:pls':
##
##     R2

# Compile cross-validation settings
set.seed(100)
myfolds <- createMultiFolds(train_p$Apps, k = 5, times = 10)
control <- trainControl("repeatedcv", index = myfolds, selectionFunction =
"oneSE")

# Train PLS model
mod1 <- train(Apps ~ ., data = train_p,
              method = "pls",
              metric = "RMSE",
              tuneLength = 20,
              trControl = control,
              preProc = c("zv","center","scale"))

summary(mod1)

## Data:    X dimension: 24 17
##  Y dimension: 24 1
## Fit method: oscorespls
## Number of components considered: 8
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X           30.84    61.11    69.40    75.50    81.94    85.84    89.89
## .outcome    82.50    87.84    92.86    95.44    96.89    97.95    98.46
##           8 comps
## X           94.22
## .outcome    98.71
```

This displays the metrics in the model including: ncom (number of predictors which is the value of M), root mean squared error, R-squared, mean absolute error etc. The lowest RMSE is preferable.

```
mod1$results

##    ncomp      RMSE  Rsquared       MAE    RMSESD RsquaredSD     MAESD
## 1      1 1084.9854 0.7133798  861.7848  521.4875  0.3403221  332.2876
## 2      2 1180.7783 0.7798500  875.6709  564.5886  0.3116584  354.9473
## 3      3 1230.5371 0.7154510  897.1544  561.0249  0.3305893  389.9395
## 4      4 1191.8641 0.7283764  905.3304  475.5401  0.2813624  375.5198
## 5      5 1132.9057 0.7626434  881.3255  443.3958  0.2641401  354.1532
## 6      6 1074.9639 0.7841007  846.2498  384.8527  0.2348362  305.2852
## 7      7 1030.9962 0.8037494  823.7655  344.5840  0.2158111  280.5208
## 8      8  977.8485 0.8270842  799.8415  320.7414  0.1946075  256.7419
## 9      9  961.4021 0.8474581  796.9018  369.8159  0.1860991  279.0550
## 10    10 1003.3505 0.8466150  833.7763  403.9196  0.1928643  310.0666
## 11    11 1066.8502 0.8373313  887.4662  421.3553  0.1989530  320.6783
## 12    12 1137.5497 0.8239245  944.5730  463.5097  0.2021272  351.6782
```

```
## 13    13 1229.7565 0.7731685 1007.7498  525.0526  0.2575168  399.2797
## 14    14 1472.4609 0.7342725 1172.4776  702.0939  0.2797465  505.0709
## 15    15 1936.8054 0.6771353 1474.1501 1218.4259  0.3113862  797.1762
## 16    16 2309.6684 0.6659932 1770.6816 1720.6389  0.3131435 1165.2617
```

(g) Comment on the results obtained. Is there much difference among the test errors resulting from these five approaches?

- There is a noticeable difference between OLS, Ridge, PCR and PLS regression in terms of mean squared error whereby Ridge regression had the lowest mean squared error followed by PLS, OLS,Lasso and then Principal Component Regression had the highest mean squared error.