Math448 HW#6

Zahraa Alshalal

2023-04-25

Conceptual Questions:

Exercise 1:

- a. Th best subset selection is the model with k predictors that has the smallest training RSS because it has the the highest correlation with the response variable. This approach can be computationally intensive for large values of p.

- b. Difficult to answer: The model with k predictors that has the smallest test RSS depends on the particular data set and cannot be determined without actually fitting and testing the models.

- c: i. True. ii. True. iii. False. iv. False. v. False.

Exercise 2:

- a. Answer iii): The lasso adds a penalty term to the least squares objective function. This penalty term restricts the flexibility of the model. As a result, the lasso may have higher bias but lower variance than least squares.
- b. Answer iii): Similar to the lasso, ridge regression introduces bias by shrinking the coefficients towards zero. However, unlike the lasso, ridge regression does not set coefficients to zero, but rather shrinks them towards zero. This results in a reduction of model complexity, making it less flexible than least squares.
- c. Answer ii): Non-linear methods are able to capture more complex relationships in the data, resulting in lower bias and improved prediction accuracy.

Applied Questions:

Exercise 9:

```
library(glmnet)
require(caret)
require(tidyverse)
```

```
library(ISLR)
sum(is.na(College))
```

```
## [1] 0
```

- a.

```
set.seed(123)
# normalize
train.size = dim(College)[1] / 2
train = sample(1:dim(College)[1], train.size)
test = (-train)
College.train = College[train, ]
College.test = College[test, ]
```

- b.

```
# Run the linear model
lm.fit = lm(Apps~., data=College.train)
lm.pred = predict(lm.fit, College.test, type="response")
linear.MSE = mean((College.test[, "Apps"] - lm.pred)^2)
cat("The test error obtained from the linear model using least squares on the training set:", linear.MS
```

```
## The test error obtained from the linear model using least squares on the training set: 1373995
```

```
(lin_info <- postResample(lm.pred, College.test$Apps))
```

```
##         RMSE     Rsquared          MAE
## 1172.1751931    0.9304721  634.5334925
```

- c.

```
train.mat = model.matrix(Apps~., data=College.train)
test.mat = model.matrix(Apps~., data=College.test)
grid = 10 ^ seq(4, -2, length=100)
mod.ridge = cv.glmnet(train.mat, College.train[, "Apps"], alpha=0, lambda=grid, thresh=1e-12)
lambda.best = mod.ridge$lambda.min
ridge.pred = predict(mod.ridge, newx=test.mat, s=lambda.best)
test_error2 = mean((College.test[, "Apps"] - ridge.pred)^2)
cat("The test error obtained from the ridge regression model using least squares on the training set:",
```

```
## The test error obtained from the ridge regression model using least squares on the training set: 143
```

```
(ridge_info <- postResample(ridge.pred, College.test$Apps))
```

```
##         RMSE     Rsquared          MAE
## 1196.4685265    0.9280182  640.2198341
```

- Test RSS is slightly higher that OLS.

- d.

```r
mod.lasso = cv.glmnet(train.mat, College.train[, "Apps"], alpha=1, lambda=grid, thresh=1e-12)
lambda.best = mod.lasso$lambda.min
lasso.pred = predict(mod.lasso, newx=test.mat, s=lambda.best)
test_error3  = mean((College.test[, "Apps"] - lasso.pred)^2)
cat("The test error obtained from the lasso model using least squares on the training set:", test_error3
```

```
## The test error obtained from the lasso model using least squares on the training set: 1397303
```

```r
(lasso_info <- postResample(lasso.pred, College.test$Apps))
```

```
##         RMSE     Rsquared          MAE
## 1182.0755195    0.9294772  620.0248921
```

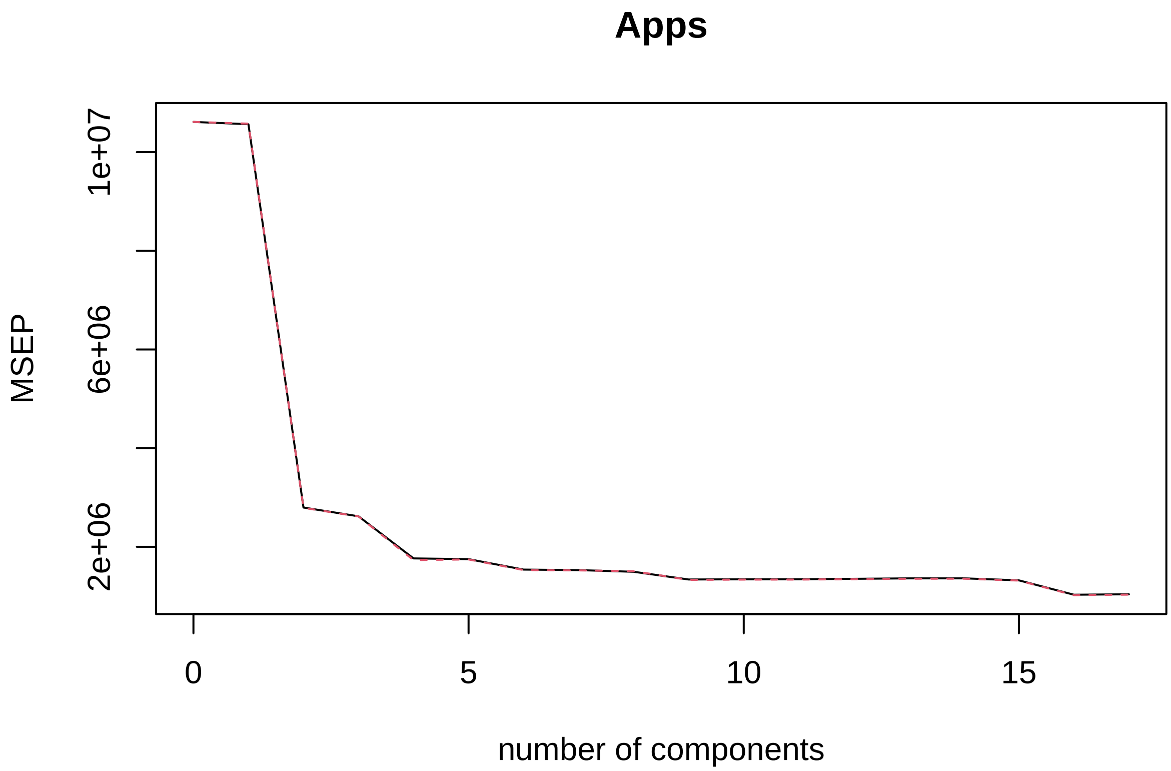
- Test RSS is slightly higher that OLS.

```r
# The coefficients
mod.lasso = glmnet(model.matrix(Apps~., data=College), College[, "Apps"], alpha=1)
predict(mod.lasso, s=lambda.best, type="coefficients")
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -6.038452e+02
## (Intercept)  .
## PrivateYes  -4.235413e+02
## Accept       1.455236e+00
## Enroll      -2.003696e-01
## Top10perc    3.367640e+01
## Top25perc   -2.403036e+00
## F.Undergrad  .
## P.Undergrad  2.086035e-02
## Outstate    -5.781855e-02
## Room.Board   1.246462e-01
## Books        .
## Personal     1.832912e-05
## PhD         -5.601313e+00
## Terminal    -3.313824e+00
## S.F.Ratio    4.478684e+00
## perc.alumni -9.796600e-01
## Expend       6.967693e-02
## Grad.Rate    5.159652e+00
```

- e.

```r
library(pls)
pcr.fit = pcr(Apps~., data=College.train, scale=T, validation="CV")
validationplot(pcr.fit, val.type="MSEP")
```



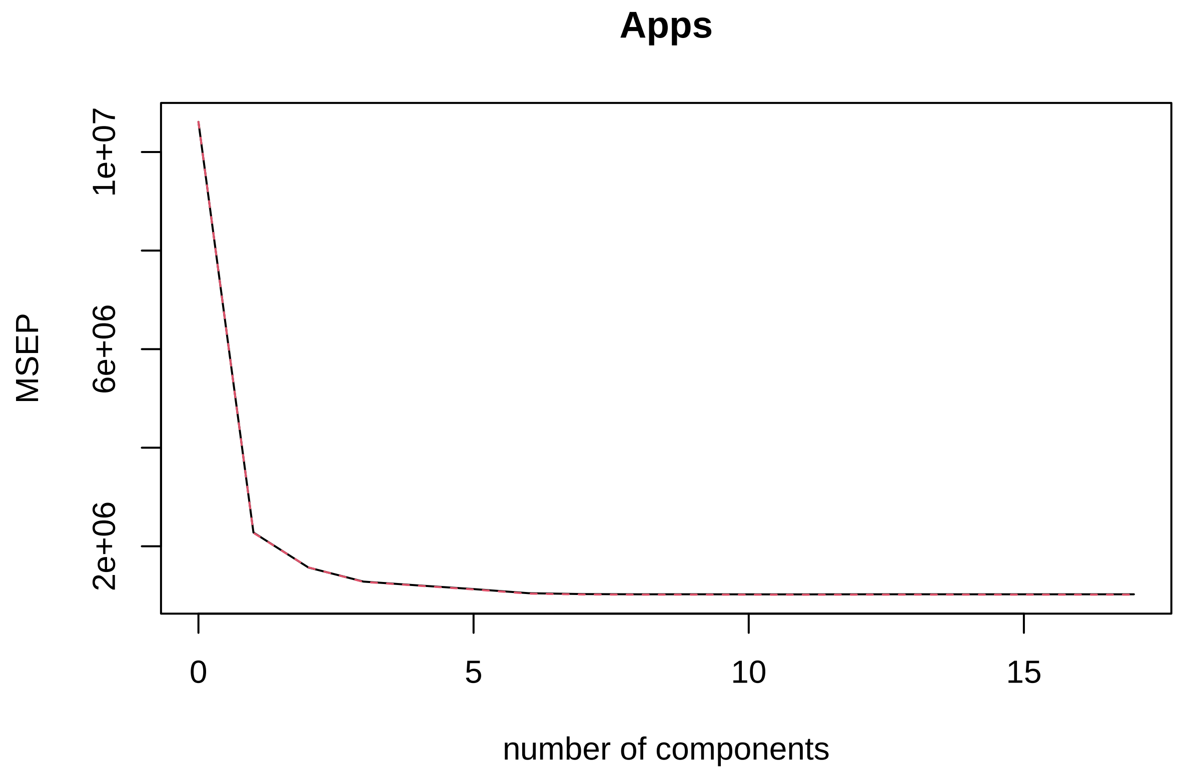
```
pcr.pred = predict(pcr.fit, College.test, ncomp=10)
test_error4 = mean((College.test[, "Apps"] - pcr.pred)^2)
cat("The test error obtained from the PCR model using least squares on the training set:", test_error3,
```

```
## The test error obtained from the PCR model using least squares on the training set: 1397303
```

```
(pcr_info <- postResample(pcr.pred, College.test$Apps))
```

```
##         RMSE     Rsquared          MAE
## 1699.2562367    0.8532601  809.2323971
```

```
pls.fit = plsr(Apps~., data=College.train, scale=T, validation="CV")
validationplot(pls.fit, val.type="MSEP")
```



```
pls.pred = predict(pls.fit, College.test, ncomp=10)
test_error5 = mean((College.test[, "Apps"] - pls.pred)^2)
cat("The test error obtained from the PLS model using least squares on the training set:", test_error5,
```

```
## The test error obtained from the PLS model using least squares on the training set: 1384151
```

```
(pls_info <- postResample(pls.pred, College.test$Apps))
```

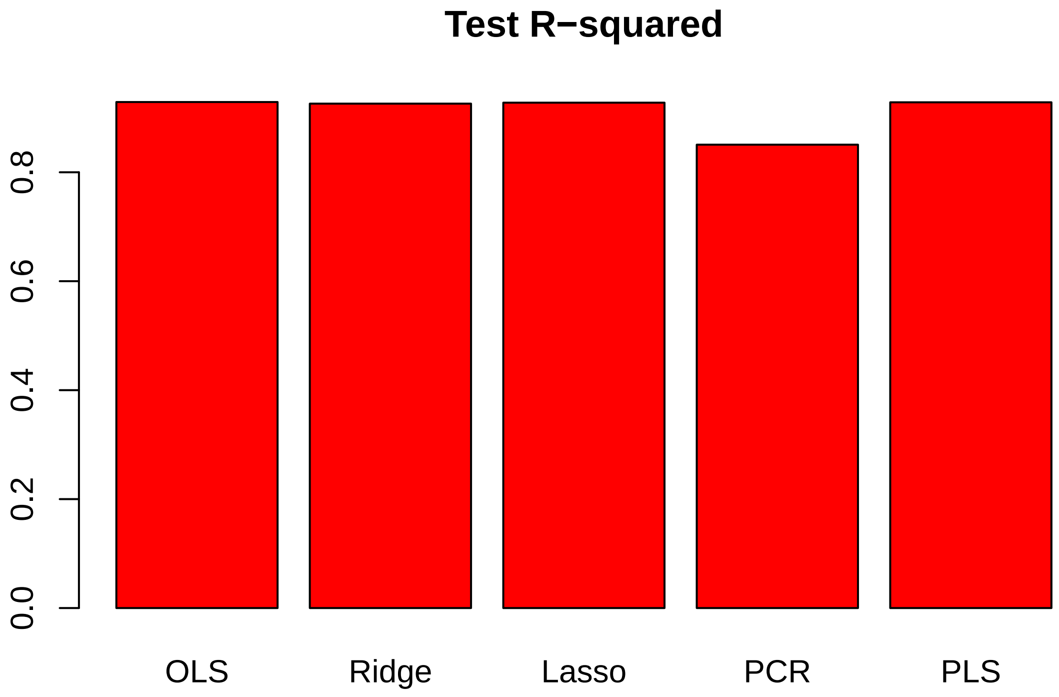
```
##         RMSE     Rsquared          MAE
## 1176.4994341    0.9299132  636.9957299
```

- g.

```
model_info <- as_data_frame(rbind(lin_info, ridge_info, lasso_info, pcr_info, pls_info))
model_info <- mutate(model_info, model = c('Linear', 'Ridge', 'Lasso', 'PCR', 'PLS'))
model_info_subset <- model_info[, c("model", "RMSE", "Rsquared")]
model_info_subset <- subset(model_info, select = c("model", "RMSE", "Rsquared"))
#resulting data frame
print(model_info_subset)
```

```
## # A tibble: 5 x 3
##   model   RMSE Rsquared
##   <chr>  <dbl>    <dbl>
## 1 Linear 1172.    0.930
## 2 Ridge  1196.    0.928
## 3 Lasso  1182.    0.929
## 4 PCR    1699.    0.853
## 5 PLS    1176.    0.930
```

```
test.avg = mean(College.test[, "Apps"])
lm.test.r2 = 1 - mean((College.test[, "Apps"] - lm.pred)^2) /mean((College.test[, "Apps"] - test.avg)^2
ridge.test.r2 = 1 - mean((College.test[, "Apps"] - ridge.pred)^2) /mean((College.test[, "Apps"] - test.
lasso.test.r2 = 1 - mean((College.test[, "Apps"] - lasso.pred)^2) /mean((College.test[, "Apps"] - test.
pcr.test.r2 = 1 - mean((College.test[, "Apps"] - pcr.pred)^2) /mean((College.test[, "Apps"] - test.avg)
pls.test.r2 = 1 - mean((College.test[, "Apps"] - pls.pred)^2) /mean((College.test[, "Apps"] - test.avg)
barplot(c(lm.test.r2, ridge.test.r2, lasso.test.r2, pcr.test.r2, pls.test.r2), col="red", names.arg=c("
```



- The plot shows that test R^2 for all models except PCR are around 0.9, with PLS having slightly higher test R^2 than others. PCR has a smaller test R^2 of around 0.8. All models except PCR predict college applications with high accuracy.