P8106 HW 1

Minjie Bao

Contents

```
library(caret)
library(ModelMetrics)
library(doBy) # which.minn()
library(RNHANES)
library(tidyverse)
library(summarytools)
library(leaps)
library(ISLR)
library(glmnet)
library(plotmo)
library(pls)

set.seed(2021)
```

Data preparation

```
train_df = read_csv('./data/solubility_train.csv') %>%
  janitor::clean_names() %>%
  na.omit()

test_df = read_csv('./data/solubility_test.csv') %>%
  janitor::clean_names() %>%
   na.omit()
```

a) linear regression method

Fit a linear model using least squares on the training data and calculate the mean squared error using the test data.

```
fit1 = lm(solubility~ ., data = train_df)
# summary(fit1)
pred.lm = predict(fit1, newdata = test_df)
MSE_linear = mean((pred.lm - test_df$solubility)^2);MSE_linear
```

```
## [1] 0.5558898
```

The mean squared error using the test data is 0.5559.

b) ridge regression model

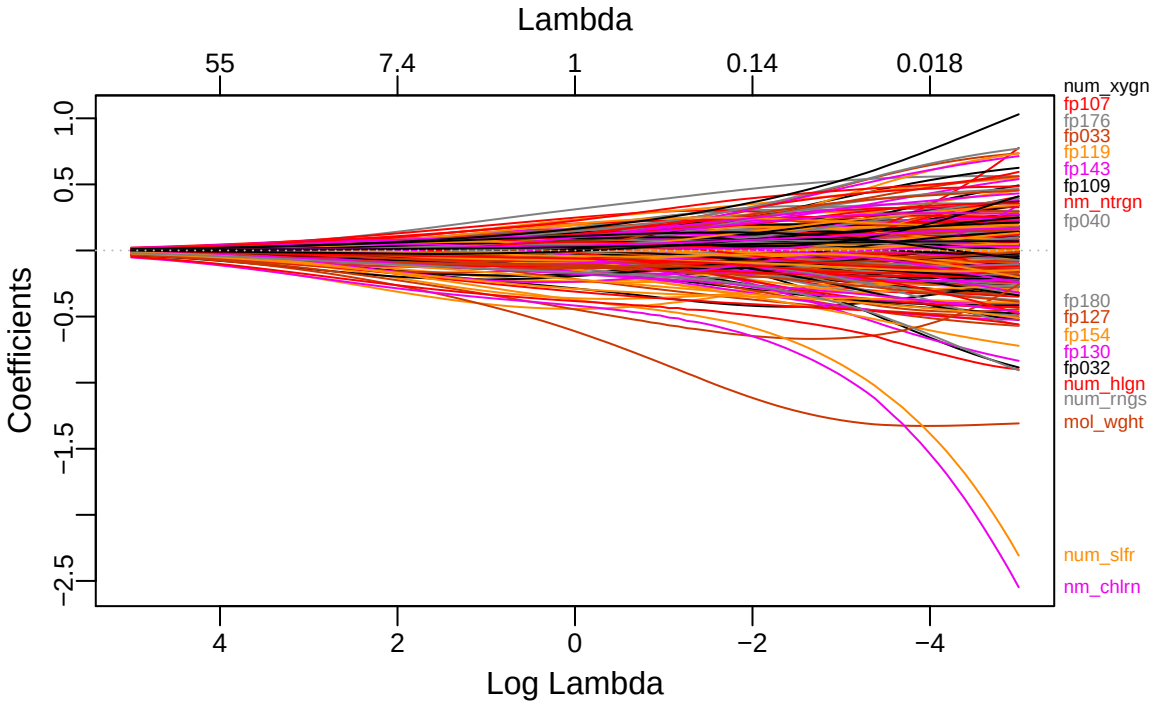
Fit a ridge regression model on the training data, with lambda chosen by cross-validation. Report the test error.

```
set.seed(1)
# fit the ridge regression (alpha = 0) with a sequence of lambdas
ridge.mod <- glmnet(x = model.matrix(solubility ~ ., train_df)[ ,-1],
```

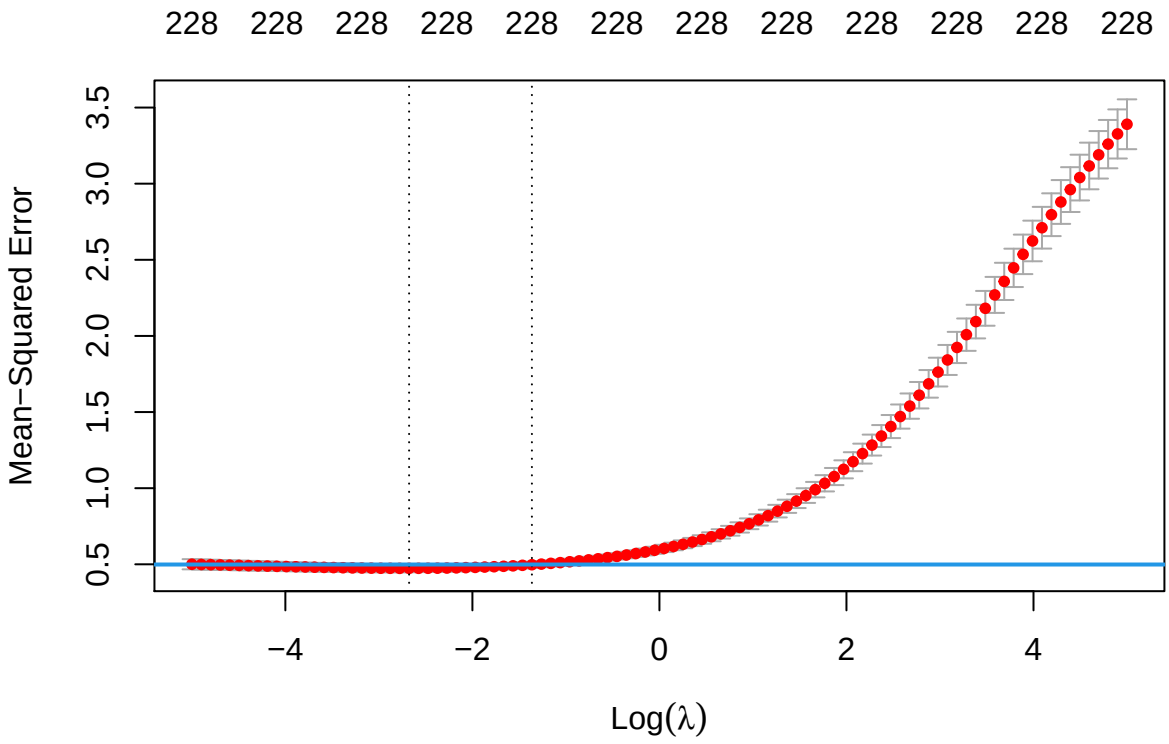
```
                    y = train_df$solubility,
                    standardize = TRUE,
                    alpha = 0,
                    lambda = exp(seq(5, -5, length = 100)))

mat.coef <- coef(ridge.mod)
dim(mat.coef)
```

```
## [1] 229 100
```

```
# Trace plot
plot_glmnet(ridge.mod, xvar = "rlambda", label = 19)
```




```
# Cross Validation
set.seed(1)
cv.ridge <- cv.glmnet(x = model.matrix(solubility ~ ., train_df)[ ,-1],
                      y = train_df$solubility,
                      type.measure = "mse",
                      alpha = 0,
                      lambda = exp(seq(5, -5, length = 100)))


plot(cv.ridge)
abline(h = (cv.ridge$cvm + cv.ridge$cvsd)[which.min(cv.ridge$cvm)], col = 4, lwd = 2)
```




```
# min CV MSE
cv.ridge$lambda.min
```

```
## [1] 0.06878513
```

```
# the 1SE rule
cv.ridge$lambda.1se
```

```
## [1] 0.2557292
```

```
# extract coefficients
pred.coeff = predict(cv.ridge, s = cv.ridge$lambda.min, type = "coefficients")

# make prediction
pred.ridge = predict(cv.ridge, newx = model.matrix(solubility ~ ., test_df)[ ,-1],
             s = "lambda.min", type = "response")

#test error
MSE_ridge = mse(test_df$solubility, pred.ridge);MSE_ridge
```

```
## [1] 0.5121469
```

For ridge model, the best lambda is 0.0688 and the mean squared error using the test data is 0.5122.

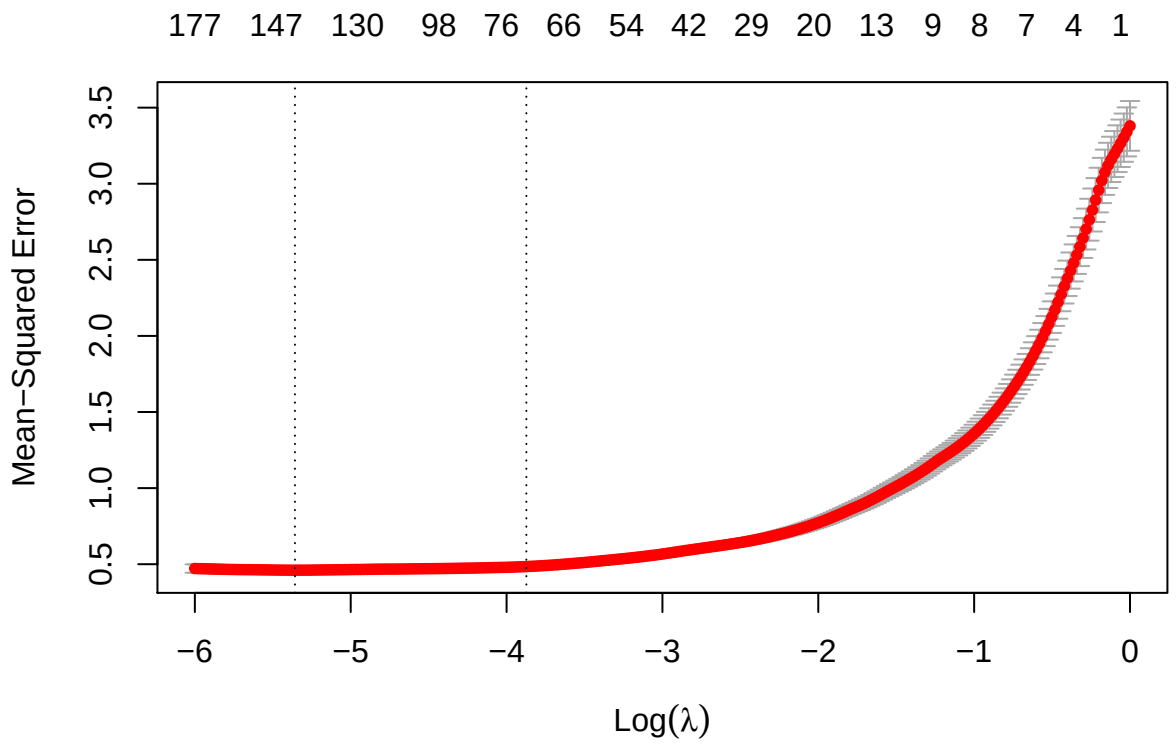
c) lasso model

Fit a lasso model on the training data, with lambda chosen by cross-validation. Report the test error and the number of non-zero coeffcient estimates in your model.

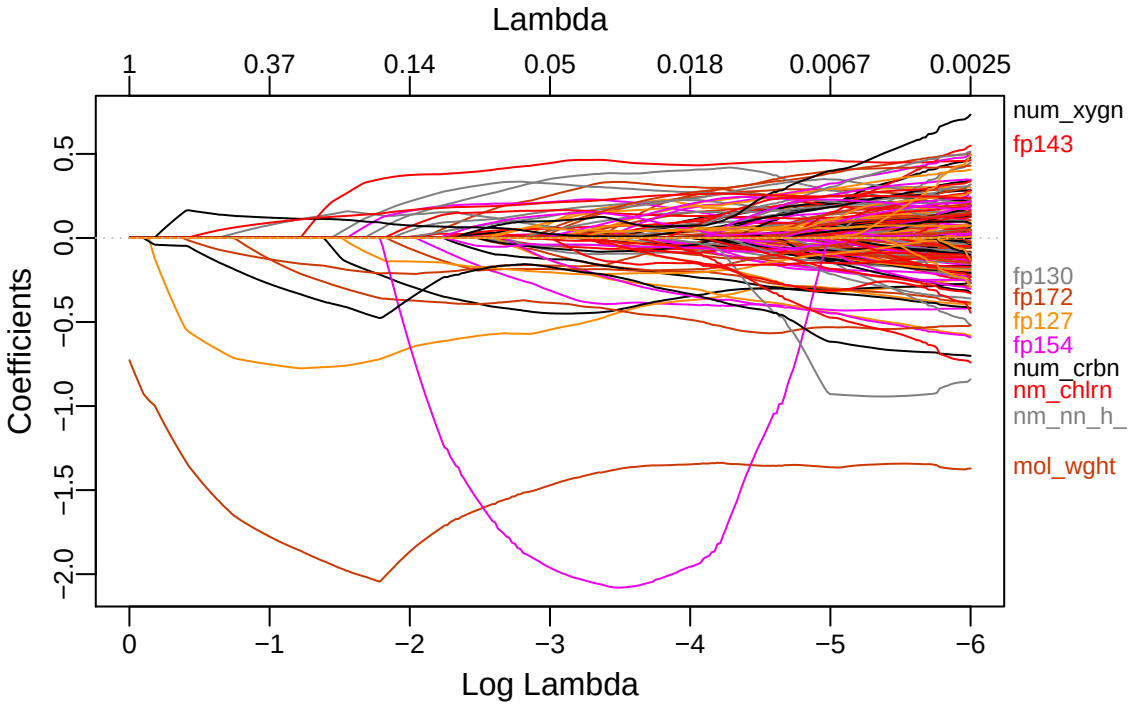
```r
#cross validation
set.seed(1)
cv.lasso <- cv.glmnet(x = model.matrix(solubility ~ ., train_df)[ ,-1],
                      y = train_df$solubility,
                      alpha = 1,
                      lambda = exp(seq(0, -6, length = 300)))

cv.lasso$lambda.min
```

```
## [1] 0.004710979
```

```r
plot(cv.lasso)
```




```r
plot_glmnet(cv.lasso$glmnet.fit)
```




```
#extract coefficient
num_coeff = sum(predict(cv.lasso, s = "lambda.min", type = "coefficients") != 0);num_coeff
```

```
## [1] 141
```

```
# make prediction
pred.lasso = predict(cv.lasso, newx = model.matrix(solubility ~ ., test_df)[ ,-1], s = "lambda.min", ty

#test error
MSE_lasso = mse(test_df$solubility, pred.lasso);MSE_lasso
```

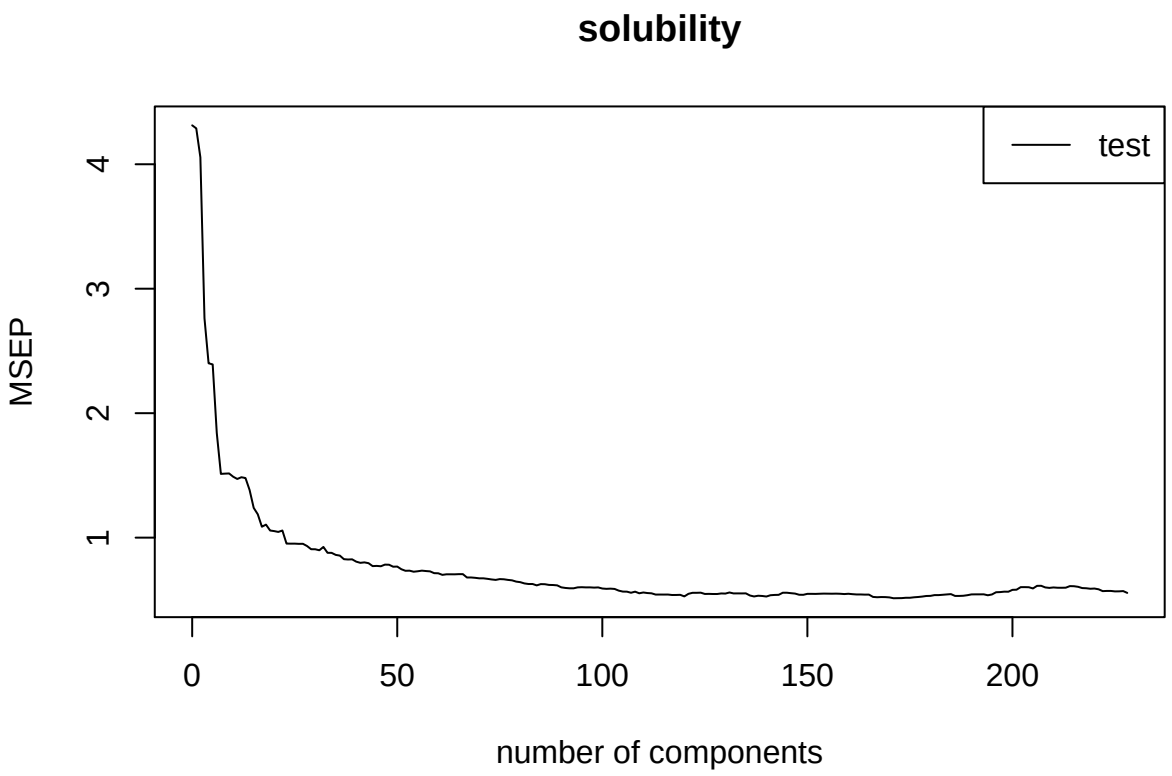
```
## [1] 0.4982291
```

For Lasso model, the best lambda is 0.0047, the mean squared error using the test data is 0.4982, and the number of non-zero coefficient estimates is 141.

d) PCR model

Fit a principle component regression model on the training data, with M chosen by cross-validation. Report the test error and the value of M selected by cross-validation.

```
set.seed(1)
pcr.mod <- pcr(solubility ~ .,
               data = train_df,
               scale = TRUE, # scale = FALSE by default
               validation = "CV")

# summary(pcr.mod)
validationplot(pcr.mod, val.type="MSEP", newdata = test_df, legendpos = "topright")
```




```
cv.mse <- RMSEP(pcr.mod)
ncomp.cv <- which.min(cv.mse$val[1,,])-1
ncomp.cv
```

```
## 152 comps
##       152
```

```
predy2.pcr <- predict(pcr.mod, newdata = test_df,
                      ncomp = ncomp.cv)
# test MSE
MSE_pcr = mean((predy2.pcr - test_df$solubility)^2)
```

For PCR model, the test error MSE using the test data is 0.5478 and the value of M selected by cross-validation is 152.

e) Model comparison

Which model will you choose for predicting solubility?

Using caret fits all the models again:

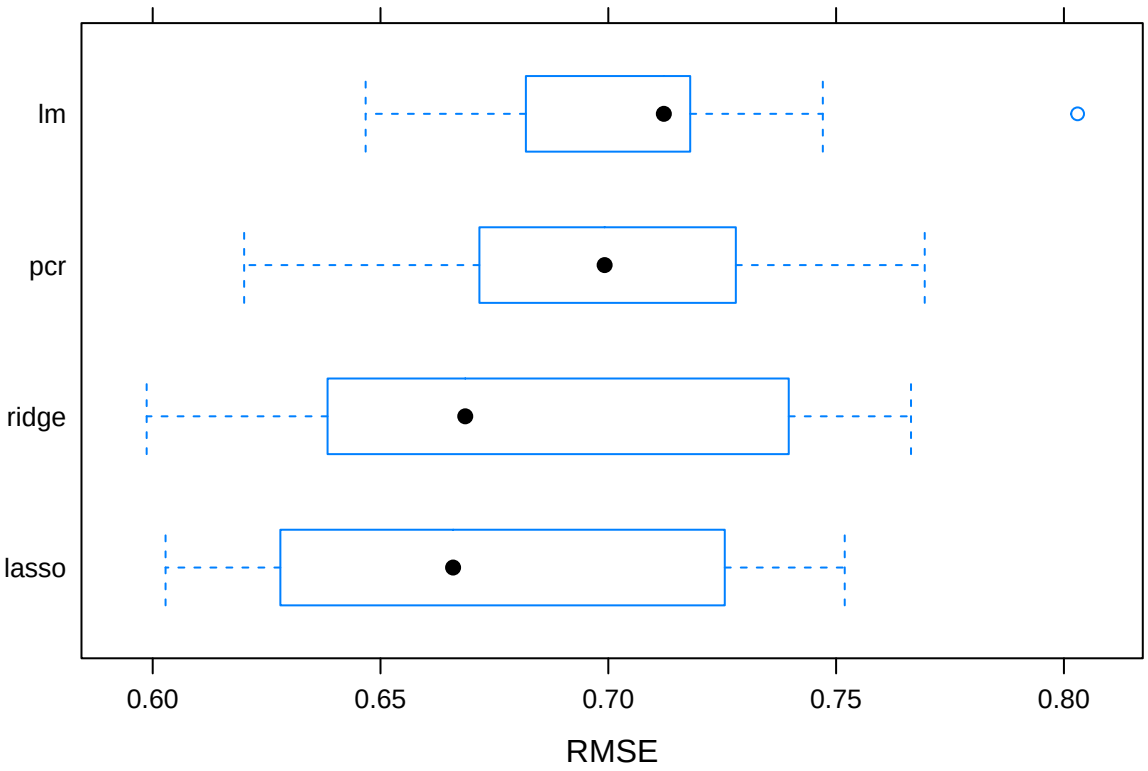
```
ctrl1 <- trainControl(method = "cv",
                      selectionFunction = "best") # "oneSE" for the 1SE rule

set.seed(1)
lm.fit <- train(x = model.matrix(solubility ~ ., train_df)[ ,-1],
```

```
                y = train_df$solubility,
                method = "lm",
                trControl = ctrl1)


set.seed(1)
ridge.fit <- train(x = model.matrix(solubility ~ ., train_df)[ ,-1],
                   y = train_df$solubility,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(5, -5, length = 100))),
                   trControl = ctrl1)


set.seed(1)
lasso.fit <- train(x = model.matrix(solubility ~ ., train_df)[ ,-1],
                   y = train_df$solubility,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(0, -6, length = 300))),
                   trControl = ctrl1)



set.seed(22)
pcr.fit <- train(x = model.matrix(solubility ~ ., train_df)[ ,-1],
                 y = train_df$solubility,
                  method = "pcr",
                  tuneLength = ncol(df),
                  trControl = ctrl1,
                  preProcess = c("center", "scale"))



set.seed(1)
resamp <- resamples(list(lm = lm.fit,
                         lasso = lasso.fit,
                         ridge = ridge.fit,
                         pcr = pcr.fit
                         ))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lm, lasso, ridge, pcr
## Number of resamples: 10
##
## MAE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm    0.4787720 0.5028170 0.5332078 0.5281167 0.5509856 0.5859704    0
## lasso 0.4739027 0.4916722 0.5072912 0.5185316 0.5511956 0.5709291    0
```

```
## ridge 0.4660475 0.4953502 0.5167305 0.5225447 0.5556487 0.5830117    0
## pcr   0.4952498 0.5225720 0.5359278 0.5383600 0.5503928 0.5870576    0
##
## RMSE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm    0.6467371 0.6850769 0.7121902 0.7080065 0.7178712 0.8030063    0
## lasso 0.6028239 0.6343875 0.6659332 0.6765232 0.7229353 0.7518872    0
## ridge 0.5986715 0.6425918 0.6686019 0.6843122 0.7365268 0.7664399    0
## pcr   0.6200718 0.6737641 0.6991964 0.6979695 0.7231468 0.7694503    0
##
## Rsquared
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm    0.8600259 0.8770213 0.8871223 0.8841123 0.8893032 0.9052887    0
## lasso 0.8692042 0.8811876 0.8927202 0.8918490 0.8996177 0.9215705    0
## ridge 0.8632437 0.8783195 0.8901298 0.8891165 0.9006366 0.9187341    0
## pcr   0.8547774 0.8747222 0.8830560 0.8850797 0.8987061 0.9110360    0
```

```
bwplot(resamp, metric = "RMSE")
```




```
cbind(c("Model", "LS", "Ridge", "Lasso", "PCR"), c("MSE", MSE_linear, MSE_ridge, MSE_lasso, MSE_pcr)) %>%
  knitr::kable()
```

Model	MSE
LS	0.555889819199859
Ridge	0.512146914044606
Lasso	0.498229081990173
PCR	0.547790475319702

From both box plot and test error (MSE) table, we can see that Lasso model has the smallest mean square error (0.4982) and linear regression model has the largest MSE (0.5559). Therefore, we conclude that Lasso model fits the data best and it is the best model for predicting solubility.