STA567 HW4

Lina Lee

10/4/2019

Model for 110m hurdles	Model Fitting Details	RMSE from LOOCV
Backward Stepwise Regression from AIC	Selected variables list: x100m, long, shotput, high	0.4322
Ridge Regression	tuning parameter value = 0.1	0.4667538
LASSO Regression	tuning parameter value = 0.2684	0.4605
Elastic Net Regression	tuning parameter value is fraction= 0.35, lambda=0.1	0.4517
Principal Component Regression	number of selected components = 4	0.4220
Partial Least Squares Regression	number of selected components = 1	0.4527
(567) Backward Stepwise Regression from RMSE	Selected variables list: x100m, long, high	0.4233

Model for 1500m run	Model Fitting Details	MSE from 5th fold CV
Backward Stepwise Regression	Selected variables list: x100m + long + shotput + x400m	12.5098
Ridge Regression	tuning parameter value = 0.03162	12.3727
LASSO Regression	tuning parameter value = 0.3105	12.2904
Elastic Net Regression	tuning parameter value = fraction=0.3 / lambda= 0	12.3041
Principal Component Regression	number of selected components = 5	12.3967
Partial Least Squares Regression	number of selected components = 2	9.7365
(567) Backward Stepwise Regression from RMSE	Selected variables list: x100m, long, shotput, x400m	12.003

```r
setwd("C:\\Users\\linal\\Desktop\\Miami2019\\STA567\\Homework\\Homework4")
load(file="Decathlons.Rdata")
head(london)

library(tidyverse)

library(caret)
```

Remove missing values

```r
london <- london %>%
  select(x110m, x1500m, x100m, long, shotput, high, x400m) %>%
```

```r
  filter(!is.na(x110m)) %>%
  filter(!is.na(x1500m))
```

(1) Backward Stepwise Regression from AIC

Model for 110m hurdles

```r
# Backward Stepwise
mod1 <- lm(x110m ~ .-x1500m,data=london)
stepBackward <- step(mod1)

## Start:  AIC=-38.93
## x110m ~ (x1500m + x100m + long + shotput + high + x400m) - x1500m
##
##             Df Sum of Sq    RSS     AIC
## - x400m     1   0.02100 3.6874 -40.782
## <none>                  3.6664 -38.931
## - shotput   1   0.39180 4.0582 -38.291
## - high      1   0.41787 4.0843 -38.125
## - long      1   0.48077 4.1472 -37.727
## - x100m     1   1.26339 4.9298 -33.233
##
## Step:  AIC=-40.78
## x110m ~ x100m + long + shotput + high
##
##             Df Sum of Sq    RSS     AIC
## <none>                  3.6874 -40.782
## - shotput   1   0.38049 4.0679 -40.229
## - long      1   0.46014 4.1476 -39.725
## - high      1   0.50006 4.1875 -39.476
## - x100m     1   2.65562 6.3430 -28.679

stepBackward

##
## Call:
## lm(formula = x110m ~ x100m + long + shotput + high, data = london)
##
## Coefficients:
## (Intercept)        x100m         long      shotput         high
##      1.4761       1.5817       0.6215      -0.1582      -3.2296

# Cross Validation
set.seed(12345)
mlr1 <- train(x110m ~ x100m + long + shotput + high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
```

```r
# RMSE for 5th-fold cross validation
min(mlr1$results$RMSE)

## [1] 0.4321686
```

Model for 1500m run

```r
# Backward Stepwise Regression from AIC
mod2 <- lm(x1500m ~ .-x110m,data=london)
stepBackward <- step(mod2)

## Start:  AIC=128.63
## x1500m ~ (x110m + x100m + long + shotput + high + x400m) - x110m
##
##             Df Sum of Sq    RSS    AIC
## - high       1     18.79 2326.5 126.84
## <none>                   2307.7 128.63
## - x100m      1    294.16 2601.9 129.75
## - long       1    332.69 2640.4 130.14
## - x400m      1    482.55 2790.3 131.57
## - shotput    1    609.47 2917.2 132.73
##
## Step:  AIC=126.84
## x1500m ~ x100m + long + shotput + x400m
##
##             Df Sum of Sq    RSS    AIC
## <none>                   2326.5 126.84
## - x100m      1    498.11 2824.6 129.89
## - x400m      1    569.98 2896.5 130.54
## - long       1    571.41 2897.9 130.56
## - shotput    1    626.86 2953.4 131.05

stepBackward

##
## Call:
## lm(formula = x1500m ~ x100m + long + shotput + x400m, data = london)
##
## Coefficients:
## (Intercept)        x100m         long      shotput        x400m
##     247.765      -25.048      -19.308        6.426        7.174

# Cross Validation
set.seed(12345)
mlr2 <- train(x1500m ~ x100m + long + shotput + x400m,
              data=london,
              method="lm",
              trControl=trainControl(method="cv",number = 5),
              preProcess = c("center", "scale"))
```

```
# RMSE for 5th-fold cross validation
min(mlr2$results$RMSE)

## [1] 12.00308
```

(2) Lasso regression

Model for 110m hurdles

```
# Set seed for reproducibility
set.seed(12345)
# Train the model
lasso_mod1<-train(x110m ~ .-x1500m ,
                  data=london,
                  method="lasso",
                  # Set up repeated k-fold cross-validation
                  trControl=trainControl(method="cv",number=5),
                  preProcess = c("center","scale"),
                  tuneLength=20)

lasso_mod1$bestTune

##    fraction
## 5 0.2684211

mean(lasso_mod1$resample$RMSE)

## [1] 0.460513
```

Model for 1500m run

```
### x1500m
set.seed(12345)
lasso_mod2<-train(x1500m ~ .-x110m ,
                  data=london,
                  method="lasso",
                  trControl=trainControl(method="cv",number=5),
                  preProcess = c("center", "scale"),
                  tuneLength=20)

lasso_mod2$bestTune

##    fraction
## 6 0.3105263

mean(lasso_mod2$resample$RMSE)

## [1] 12.29043
```

(3) Rigde regression

Model for 110m hurdles

```
set.seed(12345)
ridge_mod1 <- train(x110m ~ .-x1500m ,
                   data=london,
                   method="ridge",
                   trControl=trainControl(method="cv",number = 5),
                   preProcess = c("center", "scale"),
                   tuneLength=20)

ridge_mod1$bestTune

##    lambda
## 20    0.1

min(ridge_mod1$results$RMSE)

## [1] 0.4667538
```

Model for 1500m run

```
set.seed(12345)
ridge_mod2 <- train(x1500m ~ .-x110m ,
                    data=london,
                    method="ridge",
                    trControl=trainControl(method="cv",number = 5),
                    preProcess = c("center", "scale"),
                    tuneLength=20)


# depending on the model, criteria to choose bestTUne is different? not
RMSE??
ridge_mod2$bestTune

##        lambda
## 17 0.03162278

min(ridge_mod2$results$RMSE)

## [1] 12.37265
```

(4) Elastic net

Model for 110m hurdles

```
set.seed(12345)
enet_mod1 <- train(x110m ~ .-x1500m ,
                  data=london,
                  method="enet",
```

```r
                  trControl=trainControl(method="cv",number = 5),
                  preProcess = c("center", "scale"),
                  tuneLength=20)

enet_mod1$bestTune

##     fraction lambda
## 387     0.35    0.1

min(enet_mod1$results$RMSE)

## [1] 0.4517055
```

Model for 1500m run

```r
set.seed(12345)
enet_mod2 <- train(x1500m ~ .-x110m ,
                   data=london,
                   method="enet",
                   trControl=trainControl(method="cv",number = 5),
                   preProcess = c("center", "scale"),
                   tuneLength=20)

enet_mod2$bestTune

##   fraction lambda
## 6      0.3      0

min(enet_mod2$results$RMSE)

## [1] 12.30414
```

(5) Principal Component Regression

Model for 110m hurdles

```r
set.seed(12345)
pcr_mod1 <- train(x110m ~ .-x1500m,
                 data=london,
                 method="pcr",
                 preProcess=c("center","scale"),
                 trControl = trainControl(method="cv",number = 5),
                 tuneGrid = data.frame(ncomp=1:6))


pcr_mod1

## Principal Component Analysis
##
## 26 samples
##  6 predictor
```

```
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 20, 22, 20, 22, 20
## Resampling results across tuning parameters:
##
##   ncomp  RMSE       Rsquared   MAE
##   1      0.4534019  0.3626028  0.3725647
##   2      0.4493509  0.4479443  0.3762921
##   3      0.4534782  0.4427727  0.3819616
##   4      0.4220778  0.4592301  0.3521685
##   5      0.4974491  0.2602942  0.4362889
##   6      0.4974491  0.2602942  0.4362889
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 4.
```

```
pcr_mod1$bestTune
```

```
##   ncomp
## 4     4
```

```
min(pcr_mod1$results$RMSE)
```

```
## [1] 0.4220778
```

Model for 1500m run

```
set.seed(12345)
pcr_mod2 <- train(x1500m ~ .-x110m,
                  data=london,
                  method="pcr",
                  preProcess=c("center","scale"),
                  trControl = trainControl(method="cv",number = 5),
                  tuneGrid = data.frame(ncomp=1:6))

pcr_mod2$bestTune
```

```
##   ncomp
## 5     5
```

```
min(pcr_mod2$results$RMSE)
```

```
## [1] 12.3967
```

(6) Partial Least Squares Regression

Model for 110m hurdles

```
set.seed(12345)
plsr_mod1 <- train(x110m ~ .-x1500m,
```

```r
                   data=london,
                   method="pls",
                   preProcess=c("center","scale"),
                   trControl = trainControl(method="cv"),
                   tuneGrid = data.frame(ncomp=1:6))


plsr_mod1$bestTune

##   ncomp
## 1     1

min(plsr_mod1$results$RMSE)

## [1] 0.4527919
```

Model for 1500m run

```r
set.seed(12345)
plsr_mod1 <- train(x1500m ~ .-x110m,
                   data=london,
                   method="pls",
                   preProcess=c("center","scale"),
                   trControl = trainControl(method="cv"),
                   tuneGrid = data.frame(ncomp=1:6))


plsr_mod1$bestTune

##   ncomp
## 2     2

min(plsr_mod1$results$RMSE)

## [1] 9.736572
```

(7) Backward Stepwise Regression from RMSE

full model

```r
set.seed(12345)
mod <- train(x110m ~ x100m+ long+ shotput+ high+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
min(mod$results$RMSE)

## [1] 0.4974491
```

STEP1

Drop one variable from the full model

```
set.seed(12345)
mod1 <- train(x110m ~ x100m+ long+ shotput+ high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE1<-min(mod1$results$RMSE)

set.seed(12345)
mod2 <- train(x110m ~ x100m+ long+ shotput+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2<-min(mod2$results$RMSE)

set.seed(12345)
mod3 <- train(x110m ~ x100m+ long+ high+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE3<-min(mod3$results$RMSE)

set.seed(12345)
mod4 <- train(x110m ~ x100m+ shotput+ high+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))

RMSE4<-min(mod4$results$RMSE)

set.seed(12345)
mod5 <- train(x110m ~ long+ shotput+ high+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE5<-min(mod5$results$RMSE)

RMSE_list<-c(RMSE1,RMSE2,RMSE3,RMSE4,RMSE5)
RMSE_list

## [1] 0.4321686 0.4732444 0.4893315 0.5027613 0.4781007
```

```r
min(RMSE_list)

## [1] 0.4321686
```

mod 1 has least RMSE 0.4322. Now our improved model is lm(x110m ~ x100m+ long+ shotput+ high)

STEP2

Drop one variable from the improved model from STEP1.

```r
set.seed(12345)
mod2_1 <- train(x110m ~ x100m+ long+ shotput,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_1<-min(mod2_1 $results$RMSE)

set.seed(12345)
mod2_2 <- train(x110m ~ x100m+ long+ high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_2<-min(mod2_2$results$RMSE)

set.seed(12345)
mod2_3 <- train(x110m ~ x100m+ shotput+ high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_3<-min(mod2_3$results$RMSE)

set.seed(12345)
mod2_4 <- train(x110m ~ long+ shotput+ high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_4<-min(mod2_4$results$RMSE)

RMSE2_list<-c(RMSE2_1,RMSE2_2,RMSE2_3,RMSE2_4)
RMSE2_list

## [1] 0.4354856 0.4233306 0.4396006 0.5540580

min(RMSE2_list)

## [1] 0.4233306
```

The second model lm(x110m ~ x100m+ long+ high) in step2 has the least RMSE as 0.4233. So, our improved model is lm(x110m ~ x100m+ long+ high).

STEP3

Drop one variable from the improved model from STEP2.

```
set.seed(12345)
mod3_1 <- train(x110m ~ x100m+ long,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE3_1<-min(mod3_1$results$RMSE)

set.seed(12345)
mod3_2 <- train(x110m ~ x100m+ high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE3_2<-min(mod3_2$results$RMSE)

set.seed(12345)
mod3_3 <- train(x110m ~long+ high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE3_3<-min(mod3_3$results$RMSE)

RMSE3_list<-c(RMSE3_1,RMSE3_2,RMSE3_3)
RMSE3_list

## [1] 0.4268190 0.4305779 0.5550041

min(RMSE3_list)

## [1] 0.426819
```

All of the model in step3 has larger RMSE than the RMSE of the final model in step2,(lm(x110m ~ x100m+ long+ high)). Therefore, our final model is lm(x110m ~ x100m+ long+ high), and RMSE is 0.4233.

1500m

full model

```
set.seed(12345)
mod <- train(x1500m ~ x100m+ long+ shotput+ high+ x400m,
             data=london,
             method="lm",
```

```
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
min(mod$results$RMSE)

## [1] 12.3967
```

STEP1

Drop one variable from the full model

```
set.seed(12345)
mod1 <- train(x1500m ~ x100m+ long+ shotput+ high,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE1<-min(mod1$results$RMSE)

set.seed(12345)
mod2 <- train(x1500m ~ x100m+ long+ shotput+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2<-min(mod2$results$RMSE)

set.seed(12345)
mod3 <- train(x1500m ~ x100m+ long+ high+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE3<-min(mod3$results$RMSE)

set.seed(12345)
mod4 <- train(x1500m ~ x100m+ shotput+ high+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))

RMSE4<-min(mod4$results$RMSE)

set.seed(12345)
mod5 <- train(x1500m ~ long+ shotput+ high+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE5<-min(mod5$results$RMSE)
```

```
RMSE_list<-c(RMSE1,RMSE2,RMSE3,RMSE4,RMSE5)
RMSE_list

## [1] 12.90815 12.00308 12.65103 13.06753 12.91901

min(RMSE_list)

## [1] 12.00308
```

mod 2 has least RMSE 12.003. Now our improved model is lm(x1500m ~ x100m+ long+ shotput+ x400m)

STEP2

Drop one variable from the improved model from STEP1.

```
set.seed(12345)
mod2_1 <- train(x1500m ~ x100m+ long+ shotput,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_1<-min(mod2_1 $results$RMSE)

set.seed(12345)
mod2_2 <- train(x1500m ~ x100m+ long+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_2<-min(mod2_2$results$RMSE)

set.seed(12345)
mod2_3 <- train(x1500m ~ x100m+ shotput+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_3<-min(mod2_3$results$RMSE)

set.seed(12345)
mod2_4 <- train(x1500m ~ long+ shotput+ x400m,
             data=london,
             method="lm",
             trControl=trainControl(method="cv",number = 5),
             preProcess = c("center", "scale"))
RMSE2_4<-min(mod2_4$results$RMSE)
```

```
RMSE2_list<-c(RMSE2_1,RMSE2_2,RMSE2_3,RMSE2_4)
RMSE2_list

## [1] 12.16900 12.42921 12.63966 12.85012

min(RMSE2_list)

## [1] 12.169
```

All of the model in step2 has larger RMSE than the RMSE of the first model(lm(x1500m ~ x100m+ long+ shotput+ x400m)). Therefore, our final model is lm(x1500m ~ x100m+ long+ shotput+ x400m), and RMSE is 12.003