**HW5 (\*이 워드 파일에 R code와 output을 기입하여 제출해주시기 바랍니다.)**

1. iris 데이터를 70 대 30으로 random split 하고 Sepal.Width의 Ridge regression의 training set 내에서 lambda min을 cv.glmnet(10-fold cv)를 통해 찾고, lambda min을 사용하여 적합시킨 Ridge regression으로training mse와 test mse를 구하시오 **(\*데이터 분리할 때, cv.glmnet 시행시에 set.seed(1) 사용; Sepal.Width 제외한 모든 predictor 사용)**

|  |  |
| --- | --- |
|  | **Data preprocessing + Data split** |
|  | |
| **10fold cv, cv.glmnet을 이용해 MSE를 최소로 갖는 lambda parameter의 값은 0.019임을 알 수 있다.** | |
|  | |
| **위에서 구한 lambda 값과 전체 training data를 이용하여 다시 model을 적합한 뒤,**  **그 model을 이용해 training MSE와 test MSE를 구한다.(for both, MSE=0.08)** | |

2. iris 데이터를 70 대 30으로 random split 하고 Sepal.Width의 Lasso regression의 training set 내에서 lambda min을 cv.glmnet(10-fold cv)를 통해 찾고, lambda min을 사용하여 적합시킨 Lasso regression으로, training mse와 test mse 그리고 선택된 변수조합을 제시하시오. **(\*데이터 분리할 때, cv.glmnet 시행시에 set.seed(1) 사용; Sepal.Width 제외한 모든 predictor 사용)**

|  |
| --- |
|  |
| **10fold cv, cv.glmnet을 이용해 MSE를 최소로 갖는 lambda parameter의 값은 0.00021임을 알 수 있다.** |
|  |
| **위에서 구한 lambda 값과 전체 training data를 이용하여 다시 model을 적합한 뒤,**  **그 model을 이용해 training MSE와 test MSE를 구한다.(for both, MSE=0.08)**    **위 결과로부터 lambda parameter에 따른 variable selection 효과는 없이 Sepal.Width를 제외한 모든 변수가 explantory variable로 선택되었음을 알 수 있다.** |

**R code**

**install.packages("leaps")**

**#install.packages("glmnet")**

**#install.packages("ISLR")**

**library(ISLR)**

**library(leaps)**

**library(glmnet)**

**getwd()**

**setwd("C:/Users/DaeHyun/Desktop/Study/●Datamining")**

**#preprocessing**

**iris<-read.csv("iris.csv")**

**dim(iris)**

**str(iris)**

**iris$Species<-factor(iris$Species)**

**str(iris)**

**sum(is.na(iris$Sepal.Width))**

**#test train split**

**set.seed(1)**

**smp\_size <- floor(0.7 \* nrow(iris))**

**train\_ind<-sample(seq\_len(nrow(iris)),size=smp\_size)**

**train <- iris[train\_ind, ]**

**test <- iris[-train\_ind, ]**

**####Ridge Regression alpha=0####**

**#lambda grid**

**dim(model.matrix(Sepal.Width~.,train))**

**grid=10^seq(10,-2,length=100)**

**ridge.mod=glmnet(model.matrix(Sepal.Width~.,train),train$Sepal.Width,**

**alpha=0,lambda=grid)**

**#default 10-fold**

**set.seed(1)**

**cv.out=cv.glmnet(model.matrix(Sepal.Width~.,train),**

**train$Sepal.Width,alpha=0)**

**plot(cv.out)**

**(bestlam=cv.out$lambda.min)**

**#after determining lambda, conduct ridge regression on entire train data**

**ridge.pred=predict(ridge.mod,s=bestlam,**

**newx=model.matrix(Sepal.Width~.,train))**

**#training MSE using tuning parameter=bestlam**

**mean((ridge.pred-train$Sepal.Width)^2)**

**ridge.pred=predict(ridge.mod,s=bestlam,**

**newx=model.matrix(Sepal.Width~.,test))**

**#test MSE using tuning parameter=bestlam**

**mean((ridge.pred-test$Sepal.Width)^2)**

**####Lasso Regression alpha=1####**

**#lambda grid**

**grid=10^seq(10,-2,length=100)**

**lasso.mod=glmnet(model.matrix(Sepal.Width~.,train),train$Sepal.Width,**

**alpha=1,lambda=grid)**

**#default 10-fold**

**set.seed(1)**

**cv.out=cv.glmnet(model.matrix(Sepal.Width~.,train),**

**train$Sepal.Width,alpha=1)**

**plot(cv.out)**

**(bestlam=cv.out$lambda.min)**

**#after determining lambda, conduct lasso regression**

**#on entire train data**

**lasso.pred=predict(lasso.mod,s=bestlam,**

**newx=model.matrix(Sepal.Width~.,train))**

**#training MSE using tuning parameter=bestlam**

**mean((lasso.pred-train$Sepal.Width)^2)**

**lasso.pred=predict(lasso.mod,s=bestlam,**

**newx=model.matrix(Sepal.Width~.,test))**

**#test MSE using tuning parameter=bestlam**

**mean((lasso.pred-test$Sepal.Width)^2)**

**out=glmnet(model.matrix(Sepal.Width~.,iris),iris$Sepal.Width,**

**alpha=1,lambda=grid)**

**lasso.coef=predict(out,type="coefficients",s=bestlam)[1:7,]**

**lasso.coef**