Project 04

Indrajith Wasala Mudiyanselage

Question 01

a) Linear regression model using all predictors was fitted and test MSE using LOOCV was 1.135158.

b) The best subset selection based on adjusted R^2 selected the model with 3 predictors (Flavor, Oakiness, and Region). The maximum adjusted R^2 was 0.8164362. Then Linear regression model using selected predictors was fitted and the test MSE using LOOCV was 0.8705717.

c) The forward stepwise selection based on adjusted R^2 selected the model with 3 predictors (Flavor, Oakiness, and Region). The maximum adjusted R^2 was 0.8164362. Then the best Linear regression model using selected predictors was fitted and the test MSE using LOOCV was 0.8705717.

d) The backward stepwise selection based on adjusted R^2 selected the model with 3 predictors (Flavor, Oakiness, and Region). The maximum adjusted R^2 is 0.8164362. Then the best Linear regression model using selected predictors was fitted and the test MSE using LOOCV was 0.8705717.

e) The optimal panalty parameter via LOOCV was, $\lambda = 0.3356315$. The test MSE of the ridge regression model was 0.7035384.

f) The optimal panalty parameter via LOOCV was, $\lambda = 0.1293366$. The test MSE of the ridge regression model was 0.715177.

g) The full linear regression has the highest MSE while the ridge regression model has the lowest MSE. The best subset, forward stepwise and backward stepwise selection methods select the same predictors. Therefore the coefficients for these three methods were the same. I recommend the ridge regression model as it has the lowest MSE.

Mehod	Intercept	Clarity	Aroma	Body	Flavor	Oakiness	Region2	Region3	MSE
a) Full L. regression	7.81437	0.01705	0.08901	0.07967	1.11723	-0.34644	-1.51285	0.97259	1.135158
b) Best subset select	8.1208				1.1920	-0.3183	-1.5155	1.0935	0.8705717
c) Forward stepwise	8.1208				1.1920	-0.3183	-1.5155	1.0935	0.8705717
d) Backward stepwise	8.1208				1.1920	-0.3183	-1.5155	1.0935	0.8705717
e) Ridge regression	7.59819	0.11284	0.23439	0.19994	0.83296	-0.30155	-1.32364	0.90713	0.70354
f) Lasso	7.83522	0.00000	0.00225	0.00000	1.06085	-0.11690	-1.30564	1.07299	0.715177

Table 1: Summary of the parameter estimates

Question 02

a) Logistic regression model using all predictors was fitted and test error rate using 10-fold cross validation was 0.2194822.

b) The best subset selection based on AIC selects the model with all the predictors but SkinThickness. Then logistic regression model using selected predictors was fitted and the test error rate using 10-fold cross-validation was 0.2199697.

c) The forward stepwise selection based on AIC selects the model with all the predictors but SkinThickness. Then logistic regression model using selected predictors was fitted and the test error rate using 10-fold cross-validation was 0.2199697.

d) The backward stepwise selection based on AIC selects the model with all the predictors but SkinThickness. Then logistic regression model using selected predictors was fitted and the test error rate using 10-fold cross-validation was 0.2199697.

e) The optimal panalty parameter via 10-fold cross-validation was, $\lambda = 0.02174657$. The test error rate of the ridge regression model was 0.2210.

f) The optimal panalty parameter via 10-fold cross-validation was, $\lambda = 0.004369393$. The test error rate of the lasso model was 0.2255.

g) When comparing the test error rates of all the methods, it seems that they are very close to each other (The full logistic model gives the lowest error rate). Therefore, all the models are equally good in this scenario. But, the predictor Skin thickness is insignificant in the model. Therefore I recommend all the models that do not include predictor Skin thickness (Best subset selection, Forward stepwise selection, Backward stepwise selection, and Lasso). In mini project 3, my recommendation was the KNN model with K=6. The test error for KNN was 0.1615. This value is still less than the test errors of previously recommended models in this project. Therefore, the mini-project 3 recommendation is better than the current (mini-project 4) recommendation.

Mehod	Inter.	Pregna.	Glucose	BloodP.	SkinThi.	Insulin	BMI	DPF	Age	TestErr
a) Full L. regression	-8.02645	0.12638	0.03372	-0.00964	0.00052	-0.00124	0.07755	0.887766	0.01294	0.2194822
b) Best subset select	-8.02731	0.12637	0.03368	-0.00958		-0.00121	0.07787	0.88949	0.01289	0.2199697
c) Forward stepwise	-8.02731	0.12637	0.03368	-0.00958		-0.00121	0.07787	0.88949	0.01289	0.2199697
d) Backward stepwise	-8.02731	0.12637	0.03368	-0.00958		-0.00121	0.07787	0.88949	0.01289	0.2199697
e) Ridge regression	-7.02371	0.10422	0.02799	-0.00670	0.00032	-0.00061	0.06417	0.74054	0.01506	0.221
f) Lasso	-7.65745	0.11711	0.03204	-0.00701	0.00000	-0.00083	0.07066	0.771098	0.77110	0.2255

Table 2: Summary of the parameter estimates

R codes

```
## ----setup, include=FALSE------------------------------------------------------------------------
knitr::opts_chunk$set(echo = TRUE)


## ----include=FALSE-------------------------------------------------------------------------------

###Question 01

wine<-read.table("wine.txt",header = T)
wine$Region<-as.factor(wine$Region)
str(wine)


## ----include=FALSE-------------------------------------------------------------------------------

## a)

model1<- lm(Quality~.,wine)
library(caret)

train_control <- trainControl(method="LOOCV")
fit.lm  <- train(Quality~. ,data=wine, trControl = train_control, method = "lm")
summary(fit.lm)
# Test MSE
test.MSE<-(as.numeric(fit.lm$results[2]))^2


## ----include=FALSE-------------------------------------------------------------------------------

## b)

library(leaps)

fit.full <- regsubsets(Quality~., wine, nvmax = 6)
fit.summary <- summary(fit.full)
fit.summary$adjr2
which.max(fit.summary$adjr2)

## ----include=FALSE-------------------------------------------------------------------------------
library(caret)

train_control <- trainControl(method="LOOCV")
fit.lm.b  <- train(Quality~Flavor+ Oakiness +Region ,data=wine, trControl = train_control, method = "lm")
summary(fit.lm.b)
# Test MSE
test.MSE.b<-(as.numeric(fit.lm.b$results[2]))^2
test.MSE.b


## ----include=FALSE-------------------------------------------------------------------------------
```

```r
## c)

fit.fwd = regsubsets(Quality~., data = wine, nvmax = 6, method = "forward")
fit.fwd.summary <- summary(fit.fwd)
fit.fwd.summary
fit.fwd.summary$adjr2
which.max(fit.fwd.summary$adjr2)


## ----include=FALSE--------------------------------------------------------------------------------------------
library(caret)

train_control <- trainControl(method="LOOCV")
fit.lm.c  <- train(Quality~Flavor+ Oakiness +Region ,data=wine, trControl = train_control, method = "lm")
summary(fit.lm.c)

# Test MSE
test.MSE.c<-(as.numeric(fit.lm.c$results[2]))^2
test.MSE.c


## ----include=FALSE--------------------------------------------------------------------------------------------

## d)

fit.bwd = regsubsets(Quality~., data = wine, nvmax = 6, method = "backward")
fit.bwd.summary <- summary(fit.bwd)
fit.bwd.summary
fit.bwd.summary$adjr2
which.max(fit.bwd.summary$adjr2)


## ----include=FALSE--------------------------------------------------------------------------------------------
library(caret)

train_control <- trainControl(method="LOOCV")
fit.lm.d  <- train(Quality~Flavor+ Oakiness +Region ,data=wine, trControl = train_control, method = "lm")
summary(fit.lm.d)

# Test MSE
test.MSE.d<-(as.numeric(fit.lm.c$results[2]))^2
test.MSE.d


## ----include=FALSE--------------------------------------------------------------------------------------------

## e)

library(glmnet)

y <- wine$Quality
x <- model.matrix(Quality ~ ., wine)[, -1]

#Get the  best lamda
set.seed(1)
cv.out <- cv.glmnet(x, y, alpha = 0, grouped=FALSE, nfolds =length(wine[,1]) )
bestlam <- cv.out$lambda.min
bestlam


## ----include=FALSE--------------------------------------------------------------------------------------------
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid,
    thresh = 1e-12)

# Test MSE for the best value of lambda
ridge.pred <- predict(ridge.mod, s = bestlam, newx = x)
mean((ridge.pred - y)^2)


## ---- include=FALSE-------------------------------------------------------------------------------------------
# Refit the model on the full dataset
out <- glmnet(x, y, alpha = 0)
# Get estimates for the best value of lambda
predict(out, type = "coefficients", s = bestlam)[1:8, ]


## ----include=FALSE--------------------------------------------------------------------------------------------

## f)

library(glmnet)

#Get the  best lamda
set.seed(1)
cv.out.l <- cv.glmnet(x, y, alpha = 1, grouped=FALSE, nfolds =length(wine[,1]) )
bestlam.l <- cv.out.l$lambda.min
bestlam.l


## ----include=FALSE--------------------------------------------------------------------------------------------
lasso.mod <- glmnet(x, y, alpha = 1, lambda = grid,
    thresh = 1e-12)

# Test MSE for the best value of lambda
lasso.pred <- predict(lasso.mod, s = bestlam.l, newx = x)
mean((lasso.pred - y)^2)


## ---- include=FALSE-------------------------------------------------------------------------------------------
# Refit the model on the full dataset
out.l <- glmnet(x, y, alpha = 1)
# Get estimates for the best value of lambda
predict(out.l, type = "coefficients", s = bestlam.l)[1:8, ]


## ----include=FALSE--------------------------------------------------------------------------------------------

################################################################################################################

### Question 02

diabetes<-read.csv("diabetes.csv",header = T)
diabetes$Outcome<-as.factor(diabetes$Outcome)
str(diabetes)


## ----include=FALSE--------------------------------------------------------------------------------------------

## a)

# fit full logistic model

full <- glm(Outcome ~ ., family = binomial, data = diabetes)
summary(full)
```

```r
## ----include=FALSE------------------------------------------------------------------------------------
library(caret)
# define training control
train_control.2 <- trainControl(method="cv", number=10)
# train the model
set.seed(1)
model.logistic <- train(Outcome~., data=diabetes, trControl=train_control.2, method="glm", family=binomial(link
# summarize results
test.MSE2a<-1-model.logistic$results[[2]]
test.MSE2a


## ----include=FALSE------------------------------------------------------------------------------------

## b)

library(bestglm)
# the Xy matrix needs y as the right-most variable:
bwt.Xy <-diabetes
bglm.AIC = bestglm(Xy = bwt.Xy, family = binomial, IC = "AIC",  TopModels = 1)
bglm.AIC.coeff<-bglm.AIC$BestModel[1]
bglm.AIC.coeff

## ----include=FALSE------------------------------------------------------------------------------------
library(caret)
set.seed(1)
fit.glm.2b  <- train(Outcome~.-SkinThickness.. ,data=diabetes, trControl = train_control.2, method="glm", family


# Test MSE
test.MSE.2b<-as.numeric(1-fit.glm.2b$results[2])
test.MSE.2b


## ----include=FALSE------------------------------------------------------------------------------------

## c)

bglm.fwd.AIC = bestglm(Xy = bwt.Xy, family = binomial, IC = "AIC",method = "forward",  TopModels = 1)
bglm.fwd.AIC.coeff<-bglm.fwd.AIC$BestModel[1]
bglm.fwd.AIC.coeff

library(caret)
set.seed(1)
fit.glm.2c  <- train(Outcome~.-SkinThickness.. ,data=diabetes, trControl = train_control.2, method="glm", family

# Test MSE
test.MSE.2c<-as.numeric(1-fit.glm.2c$results[2])
test.MSE.2c


## ----include=FALSE------------------------------------------------------------------------------------

## d)

bglm.bwd.AIC = bestglm(Xy = bwt.Xy, family = binomial, IC = "AIC",method = "backward",  TopModels = 1)
bglm.bwd.AIC.coeff<-bglm.bwd.AIC$BestModel[1]
bglm.bwd.AIC.coeff

library(caret)
set.seed(1)
fit.glm.2d <- train(Outcome~.-SkinThickness.. ,data=diabetes, trControl = train_control.2, method="glm", family=
# Test MSE
test.MSE.2d<-as.numeric(1-fit.glm.2d$results[2])
test.MSE.2d


## ----include=FALSE-------------------------------------------------------------------------------------------

## e)

library(glmnet)

y <- diabetes$Outcome
x <- model.matrix(Outcome ~ ., diabetes)[, -1]

#Get the  best lamda
set.seed(1)
cv.out.2 <- cv.glmnet(x, y, alpha = 0, nfolds =10, family="binomial" )
bestlam.2 <- cv.out.2$lambda.min
bestlam.2


## ----include=FALSE-------------------------------------------------------------------------------------------
grid <- 10^seq(10, -2, length = 100)
ridge.mod.2 <- glmnet(x, y, alpha = 0,family="binomial", lambda = grid, thresh = 1e-12)

# Test MSE for the best value of lambda
ridge.pred.2 <- predict(ridge.mod.2, s = bestlam.2, newx = x,type = "response")
prd<-ifelse(ridge.pred.2>=0.5 ,"1","0")

#test error rate
mean(prd!=y)


## ---- include=FALSE------------------------------------------------------------------------------------------
# Refit the model on the full dataset
out.2 <- glmnet(x, y, alpha = 0, family="binomial")
# Get estimates for the best value of lambda
predict(out.2, type = "coefficients", s = bestlam.2)[1:9, ]


## ----include=FALSE-------------------------------------------------------------------------------------------

## f)

library(glmnet)

#Get the  best lamda
set.seed(1)
cv.out.2f <- cv.glmnet(x, y, alpha = 1, nfolds =10, family="binomial" )
bestlam.2f <- cv.out.2f$lambda.min
bestlam.2f


## ----include=FALSE-------------------------------------------------------------------------------------------
grid <- 10^seq(10, -2, length = 100)
ridge.mod.2f <- glmnet(x, y, alpha = 1,family="binomial", lambda = grid, thresh = 1e-12)

# Test MSE for the best value of lambda
ridge.pred.2f <- predict(ridge.mod.2f, s = bestlam.2f, newx = x,type = "response")
prd.f<-ifelse(ridge.pred.2f>=0.5 ,"1","0")

#test error rate
mean(prd.f!=y)
```

```
## ---- include=FALSE------------------------------------------------------------------------------------------
# Refit the model on the full dataset
out.2f <- glmnet(x, y, alpha = 1, family="binomial")
# Get estimates for the best value of lambda
predict(out.2f, type = "coefficients", s = bestlam.2f)[1:9, ]
```