Stat4360_miniproject4

John Kenney

11/1/2021

Section 1
Question 1

For Q1:(a) - (f) see the table below it contains all the information for each part and each column is a part of question 1 denoted by its column name.

1(g)

Table 1: Problem1's a-f coeff and their MSE

	(1a)	(1b)	(1c)	(1d)	(1e)	(1f)
(Intercept)	7.814370	8.1208167	8.1208167	8.1208167	7.5981854	7.8352151
Clarity	0.017050	0.0000000	0.0000000	0.0000000	0.1128379	0.0000000
Aroma	0.089010	0.0000000	0.0000000	0.0000000	0.2343928	0.0022471
Body	0.079670	0.0000000	0.0000000	0.0000000	0.1999401	0.0000000
Flavor	1.117230	1.1920393	1.1920393	1.1920393	0.8329586	1.0608452
Oakiness	-0.346440	-0.3183165	-0.3183165	-0.3183165	-0.3015529	-0.1168979
Region2	-1.512850	-1.5154840	-1.5154840	-1.5154840	-1.3236383	-1.3056417
Region3	0.972590	1.0935478	1.0935478	1.0935478	0.9071328	1.0729925
MSE	1.135158	0.8705717	0.8705717	0.8705717	1.0838052	0.9984604

From each model tested in parts 1a-1f I would recommend a linear model using the best, forward, and backward subset selection models since 1b-1d have the same subset selection and the same MSE. The best model would then be a lm with the subset {(Intercept),Flavor,Oakiness,Region2,Region3}.

Question 2

For Q2:(a) - (f) see the table below it contains all the information for each part and each column is a part of question 2 denoted by its column name.

2(g)

Table 2: Problem2's a-f coeff and their test error

	(2a)	(2b)	(2c)	(2d)	(2e)	(2f)
(Intercept)	-8.0264511	-8.0273146	-8.0273146	-8.0273146	-7.5580928	-7.6574475
Pregnancies..	0.1263845	0.1263707	0.1263707	0.1263707	0.1156036	0.1171104
Glucose..	0.0337202	0.0336810	0.0336810	0.0336810	0.0309991	0.0320424
BloodPressure..	-0.0096446	-0.0095806	-0.0095806	-0.0095806	-0.0082262	-0.0070129
SkinThickness..	0.0005185	0.0000000	0.0000000	0.0000000	0.0003548	0.0000000
Insulin..	-0.0012426	-0.0012123	-0.0012123	-0.0012123	-0.0009287	-0.0008313
BMI..	0.0775549	0.0778743	0.0778743	0.0778743	0.0712562	0.0706586
DiabetesPedigreeFunction..	0.8877583	0.8894946	0.8894946	0.8894946	0.8174384	0.7710979
Age..	0.0129414	0.0128944	0.0128944	0.0128944	0.0141208	0.0117124
test_error	0.2194822	0.2194822	0.2194822	0.2194822	0.2205000	0.2205000

Based on the table above I would recommend the full logistic model and best, forward, and backward selection models because they provide me the lowest and same Miss classification rate. Even though the full logistic regression model has different coefficients I am getting the same missclassification rate which seems odd to me but this may not be uncommon I will try to figure out the reason for this instance at a later date. The same thing occurred for Ridge and Lasso having the same missclassification error even with different coefficients so will try to get to the bottom of why this occurs next week. For the time being though I will recommend the models 2a-2d. Based on the results from miniproject 3 I would recommend the use of knn the same I recommended last miniproject because I got an error rate of 0.1715. On that note I noticed on the last project I noticed we used LOOCV to calculate the error and this time we are using 10-fold cross validation so I am not sure how comparable the errors are from this miniproject and the previous one.

Section 2

```
#load in the libraries
library(reshape2)
library(ggplot2)
library(dplyr)
library(MASS)
library(lmtest)
library(car)
library(caret)
library(leaps)
library(ISLR)
library(glmnet)
library(kableExtra)
#set working directory
setwd("C:/Users/jkenn/OneDrive/Documents/R/Stat4360/miniprojects/miniproject4/")
wine = read.table("wine.txt", header = T)
wine$Region = as.factor(wine$Region)#factor categorical varaible
summary(wine)
#View(wine)
```

```
#1a
set.seed(1)
#set train using LOOCV
train_control <- trainControl(method="LOOCV")
# train the model
model_1a <- train(Quality~., data=wine, trControl=train_control, method="lm")
print(model_1a)
summary(model_1a)
#model$results
paste("MSE:",(model_1a$results[2])^2)
#make a column of coef and mse for model 1a
a <- rbind(7.81437,0.01705,0.08901,0.07967,1.11723,
           -0.34644,-1.51285,0.97259,(model_1a$results[2])^2)
rownames(a) <- c("(Intercept)","Clarity","Aroma","Body","Flavor",
                 "Oakiness","Region2","Region3","MSE")
colnames(a) <- c("(1a)")
a
```

```
# predict function the professor gives us to get coefficients of regsubset model
predict.regsubsets <- function(object, newdata, id, ...) {
    form <- as.formula(object$call[[2]])
    mat <- model.matrix(form, newdata)
    coefi <- coef(object, id = id)
    xvars <- names(coefi)
    mat[, xvars] %*% coefi
}
```

```
#1b
#find the best model selection on the basis of adjusted r^2 to preform LOOCV on
cv.mean_1b <- c()
best.fit <- regsubsets(Quality ~ ., data = wine, nvmax = ncol(wine))
n <- which.max(summary(best.fit)$adjr2)
#does LOOCV and saves MSE
```

```
for (i in 1:nrow(wine)){
  best.fit <- regsubsets(Quality ~ ., data = wine[-c(i),],
                         nvmax = ncol(wine))
  pred <-  predict(best.fit, wine[i,], id = n)
  cv.mean_1b <- c(cv.mean_1b,mean((wine[i,6] - pred)^2))
}
MSE_1b <- mean(cv.mean_1b)#calcs total MSE of LOOCV
print(MSE_1b)
#use to extract coeff
best.fit <- regsubsets(Quality ~ ., data = wine,
                       nvmax = ncol(wine))
n <- which.max(summary(best.fit)$adjr2)
coef(best.fit, n)
#save the coeff and MSe of best subset selection method
b <- rbind(8.1208167,0,0,0,1.1920393,-0.3183165,
           -1.5154840,1.0935478,MSE_1b)
rownames(b) <- c("(Intercept)","Clarity","Aroma","Body",
                 "Flavor","Oakiness","Region2","Region3","MSE")
colnames(b) <- c("(1b)")
b
```

```
#1c
#find the best forward selection model on the basis of
#adjusted r^2 to preform LOOCV on it
cv.mean_1c <- c()
best.fit.forward <- regsubsets(Quality ~ ., data = wine,
                               nvmax = ncol(wine), method = "forward")
n <- which.max(summary(best.fit.forward)$adjr2)
#does LOOCV and saves MSE using the model subset selection found above
for (i in 1:nrow(wine)){
  best.fit.forward <- regsubsets(Quality ~ ., data = wine[-c(i),],
                                 nvmax = ncol(wine), method = "forward")
  pred.forward <-  predict(best.fit.forward, wine[i,], id = n)
  cv.mean_1c <- c(cv.mean_1c,mean((wine[i,6] - pred.forward)^2))
}
#LOOCV error
MSE_1c <- mean(cv.mean_1c)
print(MSE_1c)
best.fit.forward <- regsubsets(Quality ~ ., data = wine,
                               nvmax = ncol(wine), method = "forward")
n <- which.max(summary(best.fit.forward)$adjr2)
coef(best.fit.forward, n)
#save coef and MSe
c <- rbind( 8.1208167,0,0,0,1.1920393,-0.3183165,
            -1.5154840,1.0935478,MSE_1c)
rownames(c) <- c("(Intercept)","Clarity","Aroma","Body",
                 "Flavor","Oakiness","Region2","Region3","MSE")
colnames(c) <- c("(1c)")
c
```

```
#1d
#find the best backward selection subset on the basis
#of adjusted r^2 to preform LOOCV on it
```

```
cv.mean_1d <- c()
best.fit.back <- regsubsets(Quality ~ ., data = wine,
                            nvmax = ncol(wine), method = "backward")
n <- which.max(summary(best.fit.back)$adjr2)
#Performs LOOCV on the subset selction found above
for (i in 1:nrow(wine)){
  best.fit.back <- regsubsets(Quality ~ ., data = wine[-c(i),],
                              nvmax = ncol(wine), method = "backward")
  pred.back <-  predict(best.fit.back, wine[i,], id = n)
  cv.mean_1d <- c(cv.mean_1d,mean((wine[i,6] - pred.back)^2))
}
MSE_1d <- mean(cv.mean_1d)
print(MSE_1d)
#Saves the coeff and MSe of best back subset found
best.fit.back <- regsubsets(Quality ~ ., data = wine,
                            nvmax = ncol(wine), method = "backward")
n <- which.max(summary(best.fit.back)$adjr2)
coef(best.fit.back, n)
d <- rbind( 8.1208167,0,0,0,1.1920393,-0.3183165,
            -1.5154840,1.0935478,MSE_1d)
rownames(d) <- c("(Intercept)","Clarity","Aroma","Body",
                 "Flavor","Oakiness","Region2","Region3","MSE")
colnames(d) <- c("(1d)")
d
```

```
# saves the respose and predictor variables into matrixs for use in Cv.glmnet
y <- wine$Quality
x <- model.matrix(Quality ~ ., wine)[, -1]
grid <- 10^seq(10, -2, length = 100)
#makes grid of lambda values
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
dim(coef(ridge.mod))
```

```
#1e
set.seed(1)
# performs LOOCV ridge regression
cv.out <- cv.glmnet(x, y, alpha = 0,nfolds = nrow(x))
plot(cv.out)
pos <- which(cv.out$lambda == cv.out$lambda.min)
#get position of best lambda
print(" ")
#print(pos)
#print(cv.out$lambda[pos])
bestlam <- cv.out$lambda.min
print(bestlam)
#print(cv.out$cvm)
#printes LOOCV error of best lambda
print(cv.out$cvm[pos])
#saves the coefficients and mse
predict(cv.out, type = "coefficients", s = bestlam)
e <- rbind( 7.5981854,0.1128379,0.2343928,0.1999401,
            0.8329586,-0.3015529,-1.3236383,0.9071328,cv.out$cvm[pos])
rownames(e) <- c("(Intercept)","Clarity","Aroma","Body",
```

```
                "Flavor","Oakiness","Region2","Region3","MSE")
colnames(e) <- c("(1e)")
e
```

```
#1f
#preforms LOOCV Lasso regression
set.seed(1)
cv.out <- cv.glmnet(x, y, alpha = 1,nfolds = nrow(x))
plot(cv.out)
#saves position of best lambda
pos <- which(cv.out$lambda == cv.out$lambda.min)
print(" ")
print(pos)
#mse for best lambda calc by LOOCV
print(cv.out$lambda[pos])
bestlam <- cv.out$lambda.min
print(bestlam)
#print(cv.out$cvm)
print(cv.out$cvm[pos])
#saves coeff and MSE
predict(cv.out, type = "coefficients", s = bestlam)
f <- rbind( 7.835215148,0,0.002247145,0,1.060845222,
            -0.116897883,-1.305641743,1.072992458,cv.out$cvm[pos])
rownames(f) <- c("(Intercept)","Clarity","Aroma","Body",
                 "Flavor","Oakiness","Region2","Region3","MSE")
colnames(f) <- c("(1f)")
f
```

```
#1g
#creates kable table to contain a-f's coeffient estimations and its MSE
kbl(cbind(a,b,c,d,e,f), booktabs = T,
    caption = "Problem1's a-f coeff and their MSE") %>%
  kable_styling(latex_options = c("striped", "scale_down"))
```

Question 2

```
library(bestglm)
#read in data
diabetes <- read.csv("diabetes.csv")
#View(diabetes)
```

```
#2a
# fitts full model using 10 fold cv
set.seed(1)
train_control <- trainControl(method="cv", number=10)
# train the model
model <- train(as.factor(Outcome)~., data=diabetes,
               trControl=train_control, method="glm",family=binomial())
print(model)
paste("mse:",1-model$results[2])
summary(model)
#saves estim coeff and misss calc error
a2 <- rbind(-8.0264511,0.1263845,0.0337202,-0.0096446,0.0005185,
```

```
            -0.0012426,0.0775549,0.8877583,0.0129414,1-model$results[2])
rownames(a2) <- c("(Intercept)","Pregnancies..","Glucose..",
                  "BloodPressure..","SkinThickness..","Insulin..","BMI..",
                  "DiabetesPedigreeFunction..","Age..","test_error")
colnames(a2) <- c("(2a)")
a2
```

```
#2b
set.seed(1)
#performs best subest selection for logostistic regression
best.logistic <- bestglm(Xy = diabetes,
            family = binomial,          # binomial family for logistic
            IC = "AIC",                 # Information criteria for
            method = "exhaustive")
summary(best.logistic$BestModel)
set.seed(1)
train_control <- trainControl(method="cv", number=10)
# train the model
#calc 10-fold cv of the subest found above
model <- train(as.factor(Outcome) ~. -c(SkinThickness..),
               data=diabetes, trControl=train_control, method="glm",family=binomial())
#print(model)
#saves coeff and missclass rate
paste("mse best subset selection:",1-model$results[2])
b2 <- rbind(-8.0273146,0.1263707,0.0336810,-0.0095806,0,
            -0.0012123,0.0778743,0.8894946,0.0128944,1-model$results[2])
rownames(b2) <- c("(Intercept)","Pregnancies..","Glucose..","BloodPressure..",
                  "SkinThickness..","Insulin..","BMI..","DiabetesPedigreeFunction..",
                  "Age..","test_error")
colnames(b2) <- c("(2b)")
b2
```

```
#2c
#performs forward selection using logistic regression and calcs the best subset
set.seed(1)
forward.best.logistic <- bestglm(Xy = diabetes,
            family = binomial,          # binomial family for logistic
            IC = "AIC",                 # Information criteria for
            method = "forward")
summary(forward.best.logistic$BestModel)
set.seed(1)
train_control <- trainControl(method="cv", number=10)
# train the model
modelc <- train(as.factor(Outcome) ~. -c(SkinThickness..), data=diabetes,
                trControl=train_control, method="glm",family="binomial")
print(modelc)
paste("mse forward selction:",1-modelc$results[2])
#saves the coeff estim and missclass rate
c2 <- rbind(-8.0273146,0.1263707,0.0336810,-0.0095806,0,-0.0012123,
            0.0778743,0.8894946,0.0128944,1-modelc$results[2])
rownames(c2) <- c("(Intercept)","Pregnancies..","Glucose..","BloodPressure..",
                  "SkinThickness..","Insulin..","BMI..","DiabetesPedigreeFunction..",
                  "Age..","test_error")
```

```
colnames(c2) <- c("(2c)")
c2
```

```
#2d
# performs backward subset selection to find the best subset to perform cv on
set.seed(1)
backward.best.logistic <- bestglm(Xy = diabetes,
            family = binomial,          # binomial family for logistic
            IC = "AIC",                 # Information criteria for
            method = "backward")
summary(backward.best.logistic$BestModel)
set.seed(1)
#does 10 fold cv on subset found above
train_control <- trainControl(method="cv", number=10)
# train the model
model <- train(as.factor(Outcome) ~. -c(SkinThickness..), data=diabetes,
               trControl=train_control, method="glm",family="binomial")
print(model)
paste("mse backward selction:",1-model$results[2])
#save coef estim and miss class rate
d2 <- rbind(-8.0273146,0.1263707,0.0336810,-0.0095806,0,-0.0012123,
            0.0778743,0.8894946,0.0128944,1-model$results[2])
rownames(d2) <- c("(Intercept)","Pregnancies..","Glucose..","BloodPressure..",
                  "SkinThickness..","Insulin..","BMI..","DiabetesPedigreeFunction..",
                  "Age..","test_error")
colnames(d2) <- c("(2d)")
d2
```

```
# save data as matrices for use in glmnet to perform ridge and lasso
y <- as.factor(diabetes$Outcome)
x <- model.matrix(as.factor(Outcome) ~ ., diabetes)[, -1]
#creates grid of lambda values
grid <- 10^seq(10, -5, length = 200)
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid, family = binomial)
dim(coef(ridge.mod))
```

```
#2e
set.seed(1)
#performs ridge regression and 10 fold cv
cv.out <- cv.glmnet(x, y, alpha = 0,nfolds = 10,lambda = grid,
                    family = "binomial",type.measure="class")
plot(cv.out)
#get position of best lambda
pos <- which(cv.out$lambda == cv.out$lambda.min)
print(" ")
print(pos)
#print(cv.out$lambda[pos])
bestlam <- cv.out$lambda.min
print(bestlam)
#print(cv.out$cvm)
#missclassification rate of the best lambda
print(cv.out$cvm[pos])
predict(cv.out, type = "coefficients", s = bestlam)
```

```
#saves the coef estim and miss class rate of ridge regression
e2 <- rbind(-7.5580928264,0.1156035612,0.0309991342,-0.0082262138,0.0003548210,
            -0.0009287064,0.0712562340,0.8174384288,0.0141208172,cv.out$cvm[pos])
rownames(e2) <- c("(Intercept)","Pregnancies..","Glucose..","BloodPressure..",
                  "SkinThickness..","Insulin..","BMI..","DiabetesPedigreeFunction..",
                  "Age..","test_error")
colnames(e2) <- c("(2e)")
e2
```

```
#2f
#performs 10 fold lasso regression
set.seed(1)
cv.out.lasso <- cv.glmnet(x, y, alpha = 1,nfolds = 10,
                          family = "binomial",type.measure="class")
plot(cv.out.lasso)
#saves position of best lambda
pos2 <- which(cv.out.lasso$lambda == cv.out.lasso$lambda.min)
print(" ")
print(pos2)
#print(cv.out$lambda[pos])
bestlam2 <- cv.out.lasso$lambda.min
print(bestlam2)
#print(cv.out$cvm)
#miss class error for lasso
print(cv.out.lasso$cvm[pos2])
predict(cv.out.lasso, type = "coefficients", s = bestlam2)
#saves the coeff estim and missclassification rate of lasso method
f2 <- rbind(-7.6574475233,0.1171103570,0.0320424063,-0.0070129336,0,
            -0.0008312978,0.0706586043,0.7710978886,0.0117124399,cv.out.lasso$cvm[pos2])
rownames(f2) <- c("(Intercept)","Pregnancies..","Glucose..","BloodPressure..",
                  "SkinThickness..","Insulin..","BMI..","DiabetesPedigreeFunction..",
                  "Age..","test_error")
colnames(f2) <- c("(2f)")
f2
```

```
#2g
#creates kable table to output a-f's coef and error of q2
kbl(cbind(a2,b2,c2,d2,e2,f2), booktabs = T,
    caption = "Problem2's a-f coeff and their test error") %>%
  kable_styling(latex_options = c("striped", "scale_down"))
```