HW2

Lee JongCheol

2024-11-02

```r
library(ISLR)
library(MASS)
library(e1071)
library(glmnet)
data(College)
x <- scale(College[,-1])
y <- College$Private
```

```r
set.seed(12345)
set <- sample(rep(c("tran", "vald", "test"), length=nrow(x)))
```

```r
x_tran <- x[set=="tran",]
x_val <- x[set=="vald",]
x_test <- x[set=="test",]

y_tran <- y[set=="tran"]
y_val <- y[set=="vald"]
y_test <- y[set=="test"]
```

```r
D1_func <- function(y, p) {
  -1/2*mean((y=="Yes")*log(p) + (y=="No")*log(1-p))
}
  # p: Yes

D2_func <- function(y, p) {
  m0 <- sum(y=="No")
  m1 <- sum(y=="Yes")

  pi <- p[y=="No"]
  pj <- p[y=="Yes"]

  res <- 0
  for (i in 1:length(pi)) {
    for (j in 1:length(pj)) {
      res <- res + (pi[i] > pj[j]) + 1/2*(pi[i] == pj[j])
    }
  }
  res <- 1/(m0*m1)*res
  return(res)
}
```

1.

```r
lr1 <- glm(y_tran ~ ., family="binomial", data = data.frame(x_tran, y_tran))
lda1 <- lda(y_tran ~ ., data = data.frame(x_tran, y_tran))
qda1 <- qda(y_tran ~ ., data = data.frame(x_tran, y_tran))
nb1 <- naiveBayes(x_tran, y_tran)
```

```r
lr1_z_test <- predict(lr1, data.frame(x_test))
lr1_p_test <- 1/(1+exp(-lr1_z_test))
lr1_p_test <- pmin(pmax(lr1_p_test, 1e-10), 1-(1e-10))

res1_1 <- data.frame(D1 = D1_func(y_test, lr1_p_test),
           D2 = D2_func(y_test, lr1_p_test),
           row.names = "Logistic")
```

```r
lda1_p_test <- predict(lda1, data.frame(x_test))$posterior[,2]
lda1_p_test <- pmin(pmax(lda1_p_test, 1e-10), 1-(1e-10))

res1_2 <- data.frame(D1 = D1_func(y_test, lda1_p_test),
           D2 = D2_func(y_test, lda1_p_test),
           row.names = "LDA")
```

```r
qda1_p_test <- predict(qda1, data.frame(x_test))$posterior[,2]
qda1_p_test <- pmin(pmax(qda1_p_test, 1e-10), 1-(1e-10))

res1_3 <- data.frame(D1 = D1_func(y_test, qda1_p_test),
           D2 = D2_func(y_test, qda1_p_test),
           row.names = "QDA")
```

```r
nb1_p_test <- predict(nb1, data.frame(x_test), type = "raw")[,2]
nb1_p_test <- pmin(pmax(nb1_p_test, 1e-10), 1-(1e-10))


res1_4 <- data.frame(D1 = D1_func(y_test, nb1_p_test),
           D2 = D2_func(y_test, nb1_p_test),
           row.names = "NB")
```

```r
rbind(res1_1,res1_2,res1_3,res1_4)
```

```
##                  D1         D2
## Logistic 0.11694659 0.02369281
## LDA      0.08777236 0.01953357
## QDA      0.24937420 0.02614379
## NB       0.37472069 0.03943850
```

2.

```r
lam1 <- 10^seq(-3, -1.5, length.out=100)
ridgelr2 <- glmnet(x_tran, y_tran, alpha=0, lambda=lam1, family="binomial")
```

```
ridgelr2_z_val <- predict(ridgelr2, x_val)
ridgelr2_p_val <- 1/(1+exp(-ridgelr2_z_val))
ridgelr2_p_val <- pmin(pmax(ridgelr2_p_val, 1e-10), 1-(1e-10))

D1_vec2 <- c()
D2_vec2 <- c()

for (i in 1:100) {
  D1_vec2[i] <- D1_func(y_val, ridgelr2_p_val[,i])
  D2_vec2[i] <- D2_func(y_val, ridgelr2_p_val[,i])
}

lamb_D1 <- lam1[which.min(D1_vec2)]
lamb_D2 <- lam1[which.min(D2_vec2)]
```

```
ridgelr2_p_test1 <- predict(ridgelr2, x_test,
                            type = "response")
ridgelr2_p_test1 <- pmin(pmax(ridgelr2_p_test1[,which.min(D1_vec2)], 1e-10), 1-(1e-10))

ridgelr2_p_test2 <- predict(ridgelr2, x_test,
                            type = "response")
ridgelr2_p_test2 <- pmin(pmax(ridgelr2_p_test2[,which.min(D2_vec2)], 1e-10), 1-(1e-10))

data.frame(D1 = c(D1_func(y_test, ridgelr2_p_test1),
                  D1_func(y_test, ridgelr2_p_test2)),
           D2 = c(D2_func(y_test, ridgelr2_p_test1),
                  D2_func(y_test, ridgelr2_p_test2)),
           row.names = c("RG1", "RG2"))
```

```
##             D1         D2
## RG1 0.08607640 0.01997920
## RG2 0.09042346 0.02064765
```

3.

```
lam2 <- 10^seq(-0.5, -3.5, length.out=100)

lassolr3 <- glmnet(x_tran, y_tran, alpha=1, lambda=lam2, family="binomial")
```

```
lassolr3_z_val <- predict(lassolr3, x_val)
lassolr3_p_val <- 1/(1+exp(-lassolr3_z_val))
lassolr3_p_val <- pmin(pmax(lassolr3_p_val, 1e-10), 1-(1e-10))

D1_vec3 <- c()
D2_vec3 <- c()

for (i in 1:100) {
  D1_vec3[i] <- D1_func(y_val, lassolr3_p_val[,i])
  D2_vec3[i] <- D2_func(y_val, lassolr3_p_val[,i])
}
```

```r
lamb_D1_3 <- lam2[which.min(D1_vec3)]
lamb_D2_3 <- lam2[which.min(D2_vec3)]
```

```r
lassolr3_p_test1 <- predict(lassolr3, x_test,
                            type = "response")
lassolr3_p_test1 <- pmin(pmax(lassolr3_p_test1[,which.min(D1_vec3)], 1e-10), 1-(1e-10))

lassolr3_p_test2 <- predict(lassolr3, x_test,
                            type = "response")
lassolr3_p_test2 <- pmin(pmax(lassolr3_p_test2[,which.min(D2_vec3)], 1e-10), 1-(1e-10))


data.frame(D1 = c(D1_func(y_test, lassolr3_p_test1),
                  D1_func(y_test, lassolr3_p_test2)),
           D2 = c(D2_func(y_test, lassolr3_p_test1),
                  D2_func(y_test, lassolr3_p_test2)),
           row.names = c("LS1", "LS2"))
```

```
##             D1         D2
## LS1 0.08527566 0.01990493
## LS2 0.29445049 0.01916221
```

4.

```r
nonzero1 <- lassolr3$beta[,which.min(D1_vec3)]!=0
nonzero2 <- lassolr3$beta[,which.min(D2_vec3)]!=0
```

```r
lr4_1 <- glm(y_tran ~ ., family="binomial",
             data = data.frame(x_tran[,nonzero1], y_tran))

lda4_1 <- lda(y_tran ~ ., data = data.frame(x_tran[,nonzero1], y_tran))

qda4_1 <- qda(y_tran ~ ., data = data.frame(x_tran[,nonzero1], y_tran))

nb4_1 <- naiveBayes(x_tran[,nonzero1], y_tran)


lr4_2 <- glm(y_tran ~ ., family="binomial",
             data = data.frame(x_tran[,nonzero2], y_tran))

lda4_2 <- lda(y_tran ~ ., data = data.frame(x_tran[,nonzero2], y_tran))

qda4_2 <- qda(y_tran ~ ., data = data.frame(x_tran[,nonzero2], y_tran))

nb4_2 <- naiveBayes(x_tran[,nonzero2], y_tran)
```

```r
lr4_1_z_test <- predict(lr4_1, data.frame(x_test[,nonzero1]))
lr4_1_p_test <- 1/(1+exp(-lr4_1_z_test))
lr4_1_p_test <- pmin(pmax(lr4_1_p_test, 1e-10), 1-(1e-10))
```

```
lr4_2_z_test <- predict(lr4_2, data.frame(x_test[,nonzero2]))
lr4_2_p_test <- 1/(1+exp(-lr4_2_z_test))
lr4_2_p_test <- pmin(pmax(lr4_2_p_test, 1e-10), 1-(1e-10))
```

```
lda4_1_p_test <- predict(lda4_1,
                         data.frame(x_test[,nonzero1]))$posterior[,2]
lda4_1_p_test <- pmin(pmax(lda4_1_p_test, 1e-10), 1-(1e-10))
```

```
lda4_2_p_test <- predict(lda4_2,
                         data.frame(x_test[,nonzero2]))$posterior[,2]
lda4_2_p_test <- pmin(pmax(lda4_2_p_test, 1e-10), 1-(1e-10))
```

```
qda4_1_p_test <- predict(qda4_1,
                         data.frame(x_test[,nonzero1]))$posterior[,2]
qda4_1_p_test <- pmin(pmax(qda4_1_p_test, 1e-10), 1-(1e-10))
```

```
qda4_2_p_test <- predict(qda4_2,
                         data.frame(x_test[,nonzero2]))$posterior[,2]
qda4_2_p_test <- pmin(pmax(qda4_2_p_test, 1e-10), 1-(1e-10))
```

```
nb4_1_p_test <- predict(nb4_1,
                        data.frame(x_test[,nonzero1]), type = "raw")[,2]
nb4_1_p_test <- pmin(pmax(nb4_1_p_test, 1e-10), 1-(1e-10))
```

```
nb4_2_p_test <- predict(nb4_2,
                        data.frame(x_test[,nonzero2]), type = "raw")[,2]
nb4_2_p_test <- pmin(pmax(nb4_2_p_test, 1e-10), 1-(1e-10))
```

```
data.frame(D1 = c(D1_func(y_test, lr4_1_p_test),
                  D1_func(y_test, lr4_2_p_test),
                  D1_func(y_test, lda4_1_p_test),
                  D1_func(y_test, lda4_2_p_test),
                  D1_func(y_test, qda4_1_p_test),
                  D1_func(y_test, qda4_2_p_test),
                  D1_func(y_test, nb4_1_p_test),
                  D1_func(y_test, nb4_2_p_test)),
           D2 = c(D2_func(y_test, lr4_1_p_test),
                  D2_func(y_test, lr4_2_p_test),
                  D2_func(y_test, lda4_1_p_test),
                  D2_func(y_test, lda4_2_p_test),
                  D2_func(y_test, qda4_1_p_test),
                  D2_func(y_test, qda4_2_p_test),
                  D2_func(y_test, nb4_1_p_test),
                  D2_func(y_test, nb4_2_p_test)),
           row.names = c("LR1", "LR2", "LDA1", "LDA2",
                         "QDA1", "QDA2", "NB1", "NB2"))
```

```
##              D1         D2
## LR1  0.08676483 0.02124183
## LR2  0.09186014 0.02191028
## LDA1 0.09214246 0.02101901
## LDA2 0.09062935 0.01901367
## QDA1 0.22629472 0.03019162
## QDA2 0.10883152 0.02436126
## NB1  0.25858856 0.04092395
## NB2  0.10897589 0.02384135
```

5.

```
library(caret)

D1_vec5 <- c()
D2_vec5 <- c()

for (k in 1:100) {
  kkkp <- knn3Train(x_tran, x_val, y_tran, k=k, prob = T)
  knn_p <- attr(kkkp, "prob")[,2]
  knn_p <- pmin(pmax(knn_p, 1e-10), 1-(1e-10))

  D1_vec5[k] <- D1_func(y_val, knn_p)
  D2_vec5[k] <- D2_func(y_val, knn_p)
}

k1hat <- which.min(D1_vec5)
k2hat <- which.min(D2_vec5)

knn5_1_p_test <- knn3Train(x_tran, x_test, y_tran, k=k1hat, prob = T)
knn5_1_p_test <- attr(knn5_1_p_test, "prob")[,2]
knn5_1_p_test <- pmin(pmax(knn5_1_p_test, 1e-10), 1-(1e-10))

knn5_2_p_test <- knn3Train(x_tran, x_test, y_tran, k=k2hat, prob = T)
knn5_2_p_test <- attr(knn5_2_p_test, "prob")[,2]
knn5_2_p_test <- pmin(pmax(knn5_2_p_test, 1e-10), 1-(1e-10))


data.frame(D1 = c(D1_func(y_test, knn5_1_p_test),
                  D1_func(y_test, knn5_2_p_test)),
           D2 = c(D2_func(y_test, knn5_1_p_test),
                  D2_func(y_test, knn5_2_p_test)),
           row.names = c("KNN1", "KNN2"))
```

```
##             D1         D2
## KNN1 0.1955180 0.02785205
## KNN2 0.1265584 0.02580957
```

6.

```
set.seed(111222)
set100 <- matrix(rep(c("tran", "vald", "test"),
                     length=nrow(x)), nrow(x), 100)
set100 <- apply(set100, 2, sample)
```

```r
res6_D1 <- matrix(0, nrow = 18, ncol=100)
res6_D2 <- matrix(0, nrow = 18, ncol=100)

for (i in 1:100) {
  x_tran <- x[set100[,i]=="tran",]
  x_vald <- x[set100[,i]=="vald",]
  x_test <- x[set100[,i]=="test",]

  y_tran <- y[set100[,i]=="tran"]
  y_vald <- y[set100[,i]=="vald"]
  y_test <- y[set100[,i]=="test"]

  lr1 <- glm(y_tran ~ ., family="binomial",
             data = data.frame(x_tran, y_tran))
  lda1 <- lda(y_tran ~ ., data = data.frame(x_tran, y_tran))
  qda1 <- qda(y_tran ~ ., data = data.frame(x_tran, y_tran))
  nb1 <- naiveBayes(x_tran, y_tran)
  ###########
  lr1_z_test <- predict(lr1, data.frame(x_test))
  lr1_p_test <- 1/(1+exp(-lr1_z_test))
  lr1_p_test <- pmin(pmax(lr1_p_test, 1e-10), 1-(1e-10))

  D1_lr_1 <- D1_func(y_test, lr1_p_test)
  D2_lr_1 <- D2_func(y_test, lr1_p_test)

  res6_D1[1,i] <- D1_lr_1
  res6_D2[1,i] <- D2_lr_1
  ###########
  lda1_p_test <- predict(lda1, data.frame(x_test))$posterior[,2]
  lda1_p_test <- pmin(pmax(lda1_p_test, 1e-10), 1-(1e-10))

  D1_lda_1 <- D1_func(y_test, lda1_p_test)
  D2_lda_1 <- D2_func(y_test, lda1_p_test)

  res6_D1[2,i] <- D1_lda_1
  res6_D2[2,i] <- D2_lda_1
  ###########
  qda1_p_test <- predict(qda1, data.frame(x_test))$posterior[,2]
  qda1_p_test <- pmin(pmax(qda1_p_test, 1e-10), 1-(1e-10))

  D1_qda_1 <- D1_func(y_test, qda1_p_test)
  D2_qda_1 <- D2_func(y_test, qda1_p_test)

  res6_D1[3,i] <- D1_qda_1
  res6_D2[3,i] <- D2_qda_1
  ###########
  nb1_p_test <- predict(nb1, data.frame(x_test), type = "raw")[,2]
  nb1_p_test <- pmin(pmax(nb1_p_test, 1e-10), 1-(1e-10))

  D1_nb_1 <- D1_func(y_test, nb1_p_test)
  D2_nb_1 <- D2_func(y_test, nb1_p_test)

  res6_D1[4,i] <- D1_nb_1
```

```r
res6_D2[4,i] <- D2_nb_1
###########
lam1 <- 10^seq(-3, -1.5, length.out=100)
ridgelr2 <- glmnet(x_tran, y_tran, alpha=0,
                   lambda=lam1, family="binomial")
ridgelr2_z_val <- predict(ridgelr2, x_val)
ridgelr2_p_val <- 1/(1+exp(-ridgelr2_z_val))
ridgelr2_p_val <- pmin(pmax(ridgelr2_p_val, 1e-10), 1-(1e-10))

D1_vec2 <- c()
D2_vec2 <- c()

for (j in 1:100) {
  D1_vec2[j] <- D1_func(y_val, ridgelr2_p_val[,j])
  D2_vec2[j] <- D2_func(y_val, ridgelr2_p_val[,j])
}

lamb_D1 <- lam1[which.min(D1_vec2)]
lamb_D2 <- lam1[which.min(D2_vec2)]

ridgelr2_p_test1 <- predict(ridgelr2, x_test,
                         type = "response")
ridgelr2_p_test1 <- pmin(pmax(ridgelr2_p_test1[,which.min(D1_vec2)],
                             1e-10), 1-(1e-10))

ridgelr2_p_test2 <- predict(ridgelr2, x_test,
                           type = "response")
ridgelr2_p_test2 <- pmin(pmax(ridgelr2_p_test2[,which.min(D2_vec2)],
                             1e-10), 1-(1e-10))

D1_rlr1_2 <- D1_func(y_test, ridgelr2_p_test1)
D2_rlr1_2 <- D2_func(y_test, ridgelr2_p_test1)

D1_rlr2_2 <- D1_func(y_test, ridgelr2_p_test2)
D2_rlr2_2 <- D2_func(y_test, ridgelr2_p_test2)

res6_D1[5,i] <- D1_rlr1_2
res6_D2[5,i] <- D2_rlr1_2

res6_D1[6,i] <- D1_rlr2_2
res6_D2[6,i] <- D2_rlr2_2
###########
lam2 <- 10^seq(-0.5, -3.5, length.out=100)
lassolr3 <- glmnet(x_tran, y_tran, alpha=1,
                   lambda=lam2, family="binomial")

lassolr3_z_val <- predict(lassolr3, x_val)
lassolr3_p_val <- 1/(1+exp(-lassolr3_z_val))
lassolr3_p_val <- pmin(pmax(lassolr3_p_val, 1e-10), 1-(1e-10))

D1_vec3 <- c()
D2_vec3 <- c()
```

```
for (m in 1:100) {
  D1_vec3[m] <- D1_func(y_val, lassolr3_p_val[,m])
  D2_vec3[m] <- D2_func(y_val, lassolr3_p_val[,m])
}

lamb_D1_3 <- lam2[which.min(D1_vec3)]
lamb_D2_3 <- lam2[which.min(D2_vec3)]

lassolr3_p_test1 <- predict(lassolr3, x_test,
                           type = "response")
lassolr3_p_test1 <- pmin(pmax(lassolr3_p_test1[,which.min(D1_vec3)],
                             1e-10), 1-(1e-10))

lassolr3_p_test2 <- predict(lassolr3, x_test,
                             type = "response")
lassolr3_p_test2 <- pmin(pmax(lassolr3_p_test2[,which.min(D2_vec3)],
                             1e-10), 1-(1e-10))

D1_llr1_3 <- D1_func(y_test, lassolr3_p_test1)
D2_llr1_3 <- D2_func(y_test, lassolr3_p_test1)

D1_llr2_3 <- D1_func(y_test, lassolr3_p_test2)
D2_llr2_3 <- D2_func(y_test, lassolr3_p_test2)

res6_D1[7,i] <- D1_llr1_3
res6_D2[7,i] <- D2_llr1_3

res6_D1[8,i] <- D1_llr2_3
res6_D2[8,i] <- D2_llr2_3
###########
nonzero1 <- lassolr3$beta[,which.min(D1_vec3)]!=0
nonzero2 <- lassolr3$beta[,which.min(D2_vec3)]!=0

lr4_1 <- glm(y_tran ~ ., family="binomial",
           data = data.frame(x_tran[,nonzero1], y_tran))
lda4_1 <- lda(y_tran ~ ., data = data.frame(x_tran[,nonzero1], y_tran))
qda4_1 <- qda(y_tran ~ ., data = data.frame(x_tran[,nonzero1], y_tran))
nb4_1 <- naiveBayes(x_tran[,nonzero1], y_tran)

lr4_2 <- glm(y_tran ~ ., family="binomial",
             data = data.frame(x_tran[,nonzero2], y_tran))
lda4_2 <- lda(y_tran ~ ., data = data.frame(x_tran[,nonzero2], y_tran))
qda4_2 <- qda(y_tran ~ ., data = data.frame(x_tran[,nonzero2], y_tran))
nb4_2 <- naiveBayes(x_tran[,nonzero2], y_tran)
###########
lr4_1_z_test <- predict(lr4_1, data.frame(x_test[,nonzero1]))
lr4_1_p_test <- 1/(1+exp(-lr4_1_z_test))
lr4_1_p_test <- pmin(pmax(lr4_1_p_test, 1e-10), 1-(1e-10))

D1_lr1_4 <- D1_func(y_test, lr4_1_p_test)
D2_lr1_4 <- D2_func(y_test, lr4_1_p_test)
###########
lr4_2_z_test <- predict(lr4_2, data.frame(x_test[,nonzero2]))
```

```r
lr4_2_p_test <- 1/(1+exp(-lr4_2_z_test))
lr4_2_p_test <- pmin(pmax(lr4_2_p_test, 1e-10), 1-(1e-10))

D1_lr2_4 <- D1_func(y_test, lr4_2_p_test)
D2_lr2_4 <- D2_func(y_test, lr4_2_p_test)
###########
lda4_1_p_test <- predict(lda4_1,
                         data.frame(x_test[,nonzero1]))$posterior[,2]
lda4_1_p_test <- pmin(pmax(lda4_1_p_test, 1e-10), 1-(1e-10))

D1_lda1_4 <- D1_func(y_test, lda4_1_p_test)
D2_lda1_4 <- D2_func(y_test, lda4_1_p_test)
###########
lda4_2_p_test <- predict(lda4_2,
                         data.frame(x_test[,nonzero2]))$posterior[,2]
lda4_2_p_test <- pmin(pmax(lda4_2_p_test, 1e-10), 1-(1e-10))

D1_lda2_4 <- D1_func(y_test, lda4_2_p_test)
D2_lda2_4 <- D2_func(y_test, lda4_2_p_test)
###########
qda4_1_p_test <- predict(qda4_1,
                         data.frame(x_test[,nonzero1]))$posterior[,2]
qda4_1_p_test <- pmin(pmax(qda4_1_p_test, 1e-10), 1-(1e-10))

D1_qda1_4 <- D1_func(y_test, qda4_1_p_test)
D2_qda1_4 <- D2_func(y_test, qda4_1_p_test)
###########
qda4_2_p_test <- predict(qda4_2,
                           data.frame(x_test[,nonzero2]))$posterior[,2]
qda4_2_p_test <- pmin(pmax(qda4_2_p_test, 1e-10), 1-(1e-10))

D1_qda2_4 <- D1_func(y_test, qda4_2_p_test)
D2_qda2_4 <- D2_func(y_test, qda4_2_p_test)
###########
nb4_1_p_test <- predict(nb4_1,
                        data.frame(x_test[,nonzero1]), type = "raw")[,2]
nb4_1_p_test <- pmin(pmax(nb4_1_p_test, 1e-10), 1-(1e-10))

D1_nb1_4 <- D1_func(y_test, nb4_1_p_test)
D2_nb1_4 <- D2_func(y_test, nb4_1_p_test)
###########
nb4_2_p_test <- predict(nb4_2,
                          data.frame(x_test[,nonzero2]), type = "raw")[,2]
nb4_2_p_test <- pmin(pmax(nb4_2_p_test, 1e-10), 1-(1e-10))

D1_nb2_4 <- D1_func(y_test, nb4_2_p_test)
D2_nb2_4 <- D2_func(y_test, nb4_2_p_test)
###########
res6_D1[9,i] <- D1_lr1_4
res6_D2[9,i] <- D2_lr1_4

res6_D1[10,i] <- D1_lr2_4
res6_D2[10,i] <- D2_lr2_4
```

```r
res6_D1[11,i] <- D1_lda1_4
res6_D2[11,i] <- D2_lda1_4

res6_D1[12,i] <- D1_lda2_4
res6_D2[12,i] <- D2_lda2_4

res6_D1[13,i] <- D1_qda1_4
res6_D2[13,i] <- D2_qda1_4

res6_D1[14,i] <- D1_qda2_4
res6_D2[14,i] <- D2_qda2_4

res6_D1[15,i] <- D1_nb1_4
res6_D2[15,i] <- D2_nb1_4

res6_D1[16,i] <- D1_nb2_4
res6_D2[16,i] <- D2_nb2_4
###########
D1_vec5 <- c()
D2_vec5 <- c()

for (k in 1:100) {
  kkkp <- knn3Train(x_tran, x_val, y_tran, k=k, prob = T)
  knn_p <- attr(kkkp, "prob")[,2]
  knn_p <- pmin(pmax(knn_p, 1e-10), 1-(1e-10))

  D1_vec5[k] <- D1_func(y_val, knn_p)
  D2_vec5[k] <- D2_func(y_val, knn_p)
}

k1hat <- which.min(D1_vec5)
k2hat <- which.min(D2_vec5)

knn5_1_p_test <- knn3Train(x_tran, x_test, y_tran, k=k1hat, prob = T)
knn5_1_p_test <- attr(knn5_1_p_test, "prob")[,2]
knn5_1_p_test <- pmin(pmax(knn5_1_p_test, 1e-10), 1-(1e-10))

knn5_2_p_test <- knn3Train(x_tran, x_test, y_tran, k=k2hat, prob = T)
knn5_2_p_test <- attr(knn5_2_p_test, "prob")[,2]
knn5_2_p_test <- pmin(pmax(knn5_2_p_test, 1e-10), 1-(1e-10))

D1_knn1_4 <- D1_func(y_test, knn5_1_p_test)
D2_knn1_4 <- D2_func(y_test, knn5_1_p_test)

D1_knn2_4 <- D1_func(y_test, knn5_2_p_test)
D2_knn2_4 <- D2_func(y_test, knn5_2_p_test)

res6_D1[17,i] <- D1_knn1_4
res6_D2[17,i] <- D2_knn1_4

res6_D1[18,i] <- D1_knn2_4
res6_D2[18,i] <- D2_knn2_4
###########
```

```
}
```

```
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1

## Warning: glm.fit:

## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1

## Warning: glm.fit:

## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1

## Warning: glm.fit:

## Warning: glm.fit:        0   1
## Warning: glm.fit:        0   1
## Warning: glm.fit:       0   1
## Warning: glm.fit:       0   1
## Warning: glm.fit:       0   1
## Warning: glm.fit:       0   1
## Warning: glm.fit:       0   1
## Warning: glm.fit:       0   1
```

```
res6_D1_mean <- apply(res6_D1, 1, mean)
res6_D2_mean <- apply(res6_D2, 1, mean)

res <- data.frame(D1 = res6_D1_mean,
           D2 = res6_D2_mean,
           row.names = c("LR", "LDA", "QDA", "NB", "RG1", "RG2", "LS1", "LS2",
              "LR1", "LDA1", "QDA1", "NB1", "LR2", "LDA2", "QDA2",
              "NB2", "KNN1", "KNN2"))

write.csv(res, file = "HW2_result.csv", row.names = F, na = "")
res
```

```
##              D1         D2
## LR   0.17146367 0.03437846
## LDA  0.10090822 0.02594792
## QDA  0.38882352 0.05153528
## NB   0.32296892 0.04291183
## RG1  0.09659582 0.02528367
## RG2  0.09703613 0.02411136
## LS1  0.09599526 0.02391848
## LS2  0.15705372 0.02161532
## LR1  0.11989893 0.02860701
## LDA1 0.11603248 0.02501714
## QDA1 0.09798827 0.02502198
## NB1  0.10015286 0.02284389
## LR2  0.28807956 0.04610921
## LDA2 0.22064510 0.03603922
## QDA2 0.25337197 0.04428493
## NB2  0.20702310 0.03657426
## KNN1 0.14285828 0.03100199
## KNN2 0.13695199 0.02953447
```

```
print("Winners in terms of each D1 and D2:")
```

```
## [1] "Winners in terms of each D1 and D2:"
```

```
apply(res, 2, function(t) rownames(res)[which.min(t)])
```

```
##    D1    D2
## "LS1" "LS2"
```