HW 3

Fan Bi (fb2234)

We will be predicting whether the housing price is expensive or not using the `sahp` dataset in the **r02pro** package.

You can run the following code to prepare the analysis.

```r
library(r02pro)     #INSTALL IF NECESSARY
library(tidyverse)  #INSTALL IF NECESSARY
library(MASS)
my_sahp <- sahp %>%
  na.omit() %>%
  mutate(expensive = sale_price > median(sale_price)) %>%
  dplyr::select(gar_car, liv_area, oa_qual, expensive)
my_sahp$expensive <- as.factor(my_sahp$expensive)
my_sahp_train <- my_sahp[1:100, ]
my_sahp_test <- my_sahp[-(1:100), ]
```

Please answer the following questions.

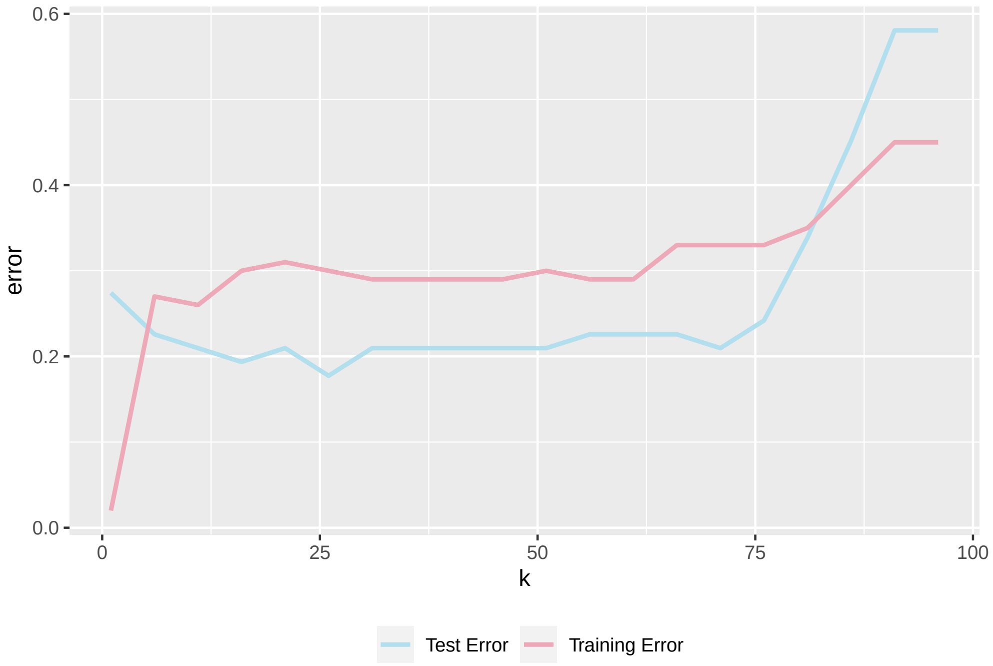
1. Use the training data `my_sahp_train` to fit a KNN model of `expensive` on variables `gar_car` and `liv_area`.

a. Vary the nearest number K from 1 to 100 with increment 5. For each K, fit the KNN classification model on the training data, and predict on the test data. Visualize the training and test error trend as a function of K. Discuss your findings.

Answer:

```r
library(caret)
trainY <- my_sahp_train$expensive
testY <- my_sahp_test$expensive
k_seq <- seq(1,100,5)

knn_train_err <- rep(0,length(k_seq))
knn_test_err <- rep(0,length(k_seq))

for(i in k_seq){
  set.seed(i)
  md <- knn3(expensive ~ gar_car + liv_area,data = my_sahp_train,k = i)
  pred_train <- predict(object = md,newdata = my_sahp_train,type = "class")
  knn_train_err[match(i,k_seq)] <- mean(trainY != pred_train)
  pred_test <- predict(object = md,newdata = my_sahp_test,type = "class")
  knn_test_err[match(i,k_seq)] <- mean(testY != pred_test)
}
```

```r
knn_dat <- data.frame(k = k_seq,train.error = knn_train_err,
                      test.error = knn_test_err) %>%
  pivot_longer(cols = train.error:test.error,names_to = "type",values_to = "error")


library(ggplot2)
ggplot(knn_dat) +
  geom_line(aes(x = k,y = error,col = type),size = 1) +
  labs(y = "error",col = "") +
  scale_color_manual(labels = c("Test Error","Training Error"),
                     values = c("lightblue2","pink2")) +
  theme(legend.position="bottom")
```



Here we set random seed before fitting KNN model each time to ensure reproducibility.

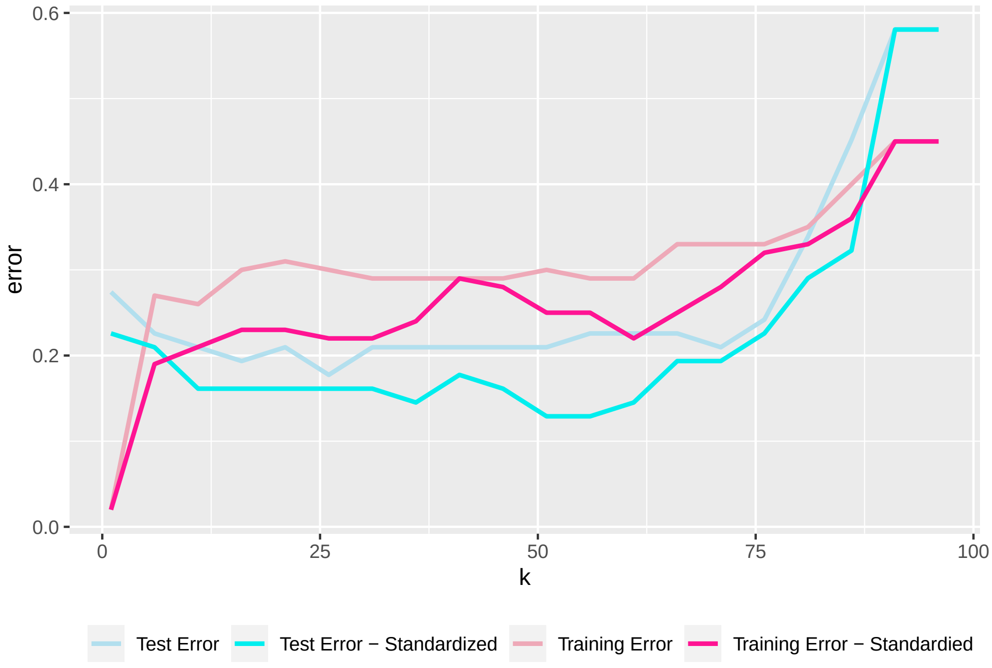
We have zero training error for binary KNN classification problem, and it will keep increasing when K increases. For test error, we obtain relatively good prediction performance when K is between 20 and 70.However, when K continuously increases,test error turns to be larger.

b. First, standardize `gar_car` and `liv_area`. Then, repeat the task in a, and visualize the training and test error together with the unstandarized version as a function of K.

Answer:

```r
knn_train_err2 <- rep(0,length(k_seq))
knn_test_err2 <- rep(0,length(k_seq))

sclform <- preProcess(my_sahp_train,method = "scale")
train_std <- predict(sclform,newdata = my_sahp_train)
```

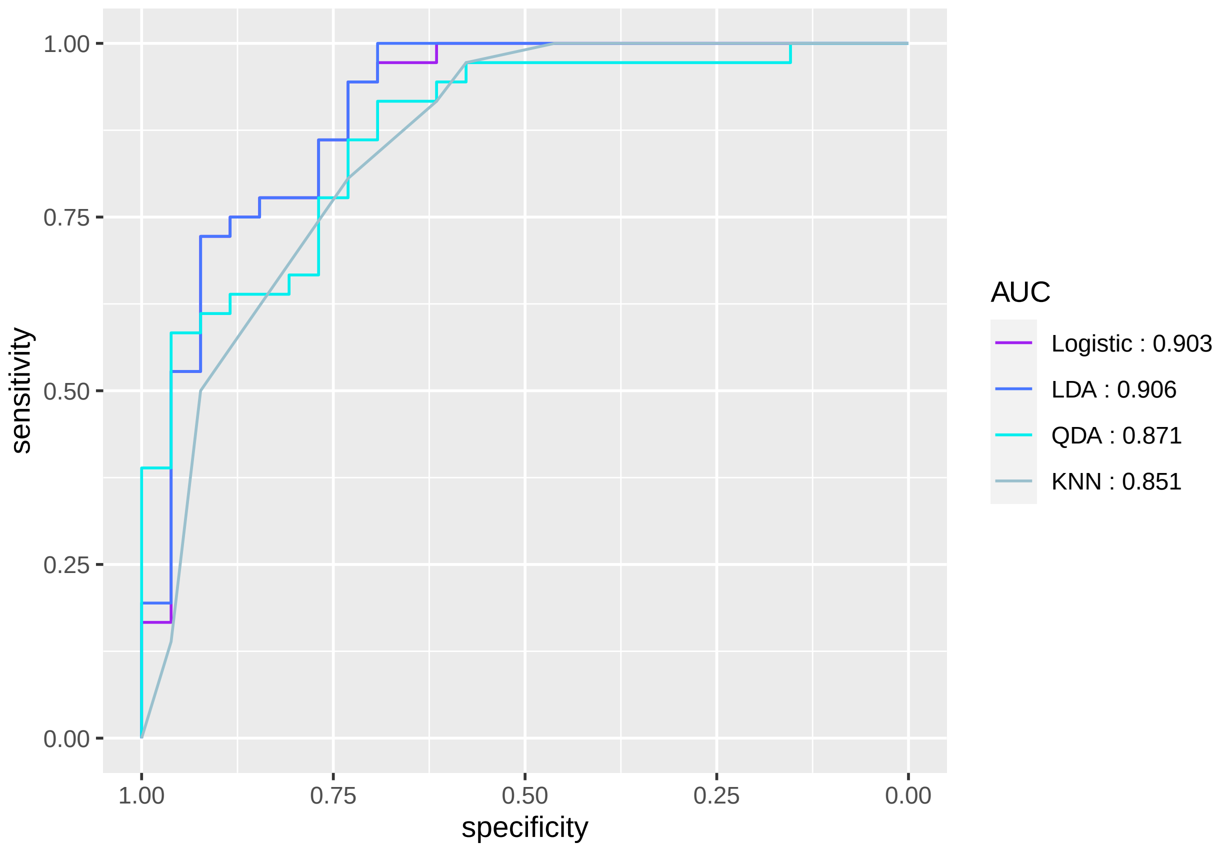
```
test_std <- predict(sclform,newdata = my_sahp_test)

for(i in k_seq){
  set.seed(i)
  md <- knn3(expensive ~ gar_car + liv_area,data = train_std,k = i)
  pred_train <- predict(object = md,newdata = train_std,type = "class")
  knn_train_err2[match(i,k_seq)] <- mean(trainY != pred_train)
  pred_test <- predict(object = md,newdata = test_std,type = "class")
  knn_test_err2[match(i,k_seq)] <- mean(testY != pred_test)
}

knn_dat2 <- data.frame(k = k_seq,train.error.std = knn_train_err2,
                       test.error.std = knn_test_err2) %>%
  pivot_longer(cols = train.error.std:test.error.std,names_to = "type",values_to = "error")


library(ggplot2)
ggplot() +
  geom_line(aes(x = k,y = error,col = type),size = 1,data = knn_dat) +
  geom_line(aes(x = k,y = error,col = type),size = 1,data = knn_dat2) +
  labs(y = "error",col = "") +
  scale_color_manual(labels = c("Test Error",
                                "Test Error - Standardized",
                                "Training Error",
                                "Training Error - Standardied"),
                     values = c("lightblue2","Cyan2","pink2","Deeppink")) +
  theme(legend.position="bottom")
```



2. Use the data `my_sahp` (without standardization) to fit four models of `expensive` on variables `gar_car` and `liv_area`, using Logistic regression, LDA, QDA, and KNN (with $K = 7$). Visualize the ROC curves for them and add the AUC values to the legend. Discuss your findings.

Answer:

```r
library(pROC)
md_name <- paste0("md_",c("Logi","LDA","QDA","KNN"))
prob_name <- paste0("pred_prob_",c("Logi","LDA","QDA","KNN"))
roc_name <- paste0("roc_",c("Logi","LDA","QDA","KNN"))
auc_name <- paste0("auc_",c("Logi","LDA","QDA","KNN"))

md_Logi <- glm(expensive ~ gar_car + liv_area, family = "binomial", data = my_sahp_train)
md_LDA <- lda(expensive ~ gar_car + liv_area , data = my_sahp_train)
md_QDA <- qda(expensive ~ gar_car + liv_area , data = my_sahp_train)
md_KNN <- knn3(expensive ~ gar_car + liv_area,data = my_sahp_train,k = 7)

pred_prob_Logi <- predict(md_Logi,newdata = my_sahp_test,type = "response")
pred_prob_LDA <- predict(md_LDA,newdata = my_sahp_test)$posterior[,2]
pred_prob_QDA <- predict(md_QDA,newdata = my_sahp_test)$posterior[,2]
pred_prob_KNN <- predict(md_KNN,newdata = my_sahp_test)[,2]

for(i in 1:length(roc_name)){
  assign(roc_name[i],roc(my_sahp_test$expensive,get(prob_name[i])))
  assign(auc_name[i], auc(get(roc_name[i])))
}

rocobj <- list(Logi = get(roc_name[1]),
               LDA = get(roc_name[2]),
               QDA = get(roc_name[3]),
               KNN = get(roc_name[4]))

auc_vec <- paste(c("Logistic", "LDA", "QDA","KNN"),":",
                 round(unname(sapply(auc_name,get)),3))


ggroc(rocobj, size = 0.5) +
  labs(col = "AUC") +
  scale_color_manual(labels = auc_vec,
                     values = c("Purple","Royalblue1",
                              "Cyan2","lightblue3"))
```



3. ISLRv2 Chapter 4 Q4

We denote on the expectation of the fraction of available observations with size p as I_p. Here we only evaluate the integral of center area and corner area.

a) Answer:

when $p = 1$, the pdf of random variable X_1 is

$$f_{X_1}(x) = 1 \quad X \in [0, 1]$$

I define the fraction function when $p = 1$ as below :

$$\omega^{(1)}(x) = \begin{cases} x + 0.05 & x \in [0, 0.05] \\ 0.1 & x \in [0.05, 0.95] \\ 1.05 - x & x \in [0.95, 1] \end{cases}$$

Then we can write the expectation of the fraction function as below :

$$
\begin{aligned}
I_1 &= \int_0^1 \omega^{(1)}(x_1) f_{X_1}(x_1) dx_1 \\
&= \int_0^{0.05} (x_1 + 0.05) dx_1 + \int_{0.05}^{0.95} 0.1 dx_1 + \int_{0.95}^{1} (1.05 - x_1) dx_1 \\
&= 2 \int_0^{0.05} (x_1 + 0.05) dx_1 + \int_{0.05}^{0.95} 0.1 dx_1 \\
&= 2 (\frac{1}{2} x_1^2 + 0.05 x_1) \Big|_0^{0.05} + 0.1 * 0.9 \\
&= 0.0075 + 0.09 = 0.0975 < 0.1
\end{aligned}
$$

b) Answer:

For $p = 2$ we have similar definition :

$$f_{X_1, X_2}(x_1, x_2) = 1 \quad X \in [0,1] \times [0,1]$$

The marginal function of fraction function when $p = 2$ is

$$
\begin{cases}
\dfrac{\partial \omega^{(2)}(x_1, x_2)}{\partial x_1} = \omega^{(1)}(x_2) \\
\dfrac{\partial \omega^{(2)}(x_1, x_2)}{\partial x_2} = \omega^{(1)}(x_1)
\end{cases}
$$

Then we can write the expectation of the fraction function as below :

$$
\begin{aligned}
I_2 &= \iint_{[0,1] \times [0,1]} \omega^{(2)}(x_1, x_2) f_{X_1, X_2}(x_1, x_2) dx_1 dx_2 \\
&= 2^2 \iint_{[0,0.05] \times [0,0.05]} (x_1 + 0.05)(x_2 + 0.05) dx_1 dx_2 + (0.1 * 0.9)^2 \\
&= (0.0075)^2 + (0.09)^2 = 0.00815625 < (0.1)^2
\end{aligned}
$$

c) Answer:

Similar to (a) and (b) we can write I_p in the form of p times integral asbelow :

$$
\begin{aligned}
I_p &= \int_{\dots} \int_{[0,1] \times \dots \times [0,1]} \omega^{(p)}(x_1, \dots, x_p) f_{X_p} dx_1 \dots dx_p \\
&= 2^p \int_{\dots} \int_{[0,0.05] \times \dots \times [0,0.05]} \prod_{i=1}^{p} (x_i + 0.05) dx_1 \dots dx_p + (0.09)^p \\
&= (0.0075)^p + (0.09)^p
\end{aligned}
$$

when $p = 100$, the expectaion of fraction function is :

$$I_{100} = (0.0075)^{100} + (0.09)^{100} < 0.1^{100}$$

d) Answer:

As the general form of expectation of the fraction of available observations used in p-dimensional KNN algorithm, we can learn the fraction will decreases exponentially when p increases, which means very few training observations are “near” any given test observation.

e) Answer :

We need to propose a new fraction function, here d is the length of each side of hypercube :

$$\omega_*^{(1)}(x) = \begin{cases} x + d/2 & x \in [0, d/2] \\ d & x \in [d/2, 1 - d/2] \\ 1 + d/2 - x & x \in [1 - d/2, 1] \end{cases}$$

For $p = 1$,

$$I_{1*} = \frac{3}{4}d^2 + \frac{9}{10}d = 10\%$$

$$\Rightarrow l_1 = d = \frac{2\sqrt{1.11}}{3} - 0.6 = 0.1024$$

For $p = 2$,

$$I_{2*} = \left(\frac{3}{4}\right)^2 d^4 + \left(\frac{9}{10}\right)^2 d^2 = 10\%$$

$$\Rightarrow l_2 = d = (l_1)^{\frac{1}{2}} = 0.3120$$

For $p = 100$,

$$I_{100*} = \left(\frac{3}{4}\right)^{100} d^{200} + \left(\frac{9}{10}\right)^{100} d^{100} = 10\%$$

$$\Rightarrow l_{100} = d = (l_1)^{\frac{1}{100}} = 0.9775$$