Johnson and Wichern (1988, Chapter 11) provide admission data for applicants to graduate schools in business. The objective is to use the GPA and GMAT scores to predict the likelihood of admission (admit, notadmit, and borderline).

Follow the steps and answer the questions:

1. Import the csv file and check the data frame of the dataset. How many students included in this data?

data <- read.csv(file.choose(), stringsAsFactor = F)

str(data)

85 students in the data frame

1. Recode the Decision variable into two levels. If decision is “admit”, De is 1, otherwise De is 0. How many students belong to class admit?   
   Hint:   
   1st way: admit$De <- factor(admit$De); levels(admit$De)[2] <- "notadmit"  
   2nd way: admit$De <- ifelse(admit$De =="admit",1,0); admit$De <- factor(admit$De)

data$De <- factor(data$De);levels(data$De)[2]<-"notadmit"

summary(data$De)

31 students are belonging to admit

1. Plot a graph between GMAT and GPA. Do students with higher GPA necessarily lead to higher GMAT scores?

data$GMAT, data$GPA, main="Scatter plot between GPA and GMAT", xlab="GMAT")

It does look like higher GPA tends to led to a high GMAT

1. Does admit class have students with higher average GPA? Use aggregate function to verify your assumption. x <- aggregate(GPA ~ De, data=admit, FUN=mean). What is the average GPA for admit class?

aggregate(data$GPA ~ data$De, FUN=mean)

Admit class has higher average GPA and the average GPA is 3.403871

1. Similarly, does admit class have students with higher average GMAT? Use aggregate function to verify your assumption. What is the average GMAT for admit class?

aggregate(data$GMAT ~ data$De, FUN=mean)

admit class have students with higher average GMAT and the average GMAT for admit class is 561.2258

1. Use prop.table to show the percentage of students belong to class admit?

prop.table(table(data$De))\*100

36.47059% are admitted

1. Look into the data. You will notice that the data is sorted by the level “admit”, ie the first set of all records are the students who got admitted. So, in order to include both levels of De in the training and testing sample, you need to randomize the sample. Inspect 10 students in the original data and randomized data.

nRows <- nrow(data)

nRows

trainSize <- ceiling(nRows \* 0.7)

trainSize

set.seed(1234)

train.index <- sample(1:nRows, trainSize, replace = F)

train.index[1:10]

data.train <- data[train.index,]

str(data.train)

str(train.index[1:10])

summary(data.train)

Original data and randomized data are completely different after creating a training size of 70% of the original data set. In addition, the admit and notadmit are mixed together.

1. What is the average GPA of students in this dataset?

summary(data.train)

The average overall GPA regardless of admit or notadmit is 2.989

1. Build a normalization function, and normalize variables GPA and GMAT. What is the average GPA after normalization?

minmax <- function(x){

ifelse(is.numeric(x),

return( ( x - min(x) ) / ( max(x) - min(x) ) ),

return (x) )

}

minmax(data.train$GPA)

minmax(data.train$GMAT)

summary(minmax(data.train$GPA))

average GPA is 0.5143

1. Create training and testing sample. Use the first 60% of students as training sample and the rest as testing sample. Round the number of training sample using floor(). How many students do you include in your testing sample?

nRows <- nrow(data)

nRows

ts <- floor(nRows\*0.6)

ts

set.seed(4567)

train <- sample(1:nRows, ts, replace = F)

train[1:10]

data.train <- data[train,]

str(data.train)

str(train[1:10])

data.test <- data[-train,]

str(data.test)

34 students are in the testing sample since the training sample contains the other 51 students.

1. Build a knn model with De as class variable to predict the admission class of testing sample using k = 5. How many students have been misclassified in total? For the misclassified students, which class do they actually belong to? Which class did we classify them?

The testing sample is from question 10

predicted.out <- knn(data.train[-3], data.test[-3], data.train$De, k=5)

CrossTable(data.test$De, predicted.out, dnn=c("Actual", "Predicted"))

7 students are misclassified

For the first column(FP) we had 3 predicted to be admitted but they are not admitted and for the second column(TN) we have predicted 4 not admitted but they are admitted

1. Use scale() to standardize the dataset from Q2. What is the average of GPA now?

data$GPA <- scale(data$GPA)

summary(data$GPA)

0 is the average now

1. Redo steps from 10 and 11.

nRows <- nrow(data)

nRows

trainSample <- floor(nRows\*0.6)

trainSample

set.seed(4567)

train <- sample(1:nRows, trainSample, replace = F)

train[1:10]

data.train <- data[train,]

str(data.train)

str(train[1:10])

data.test <- data[-train,]

str(data.test)

predicted.out <- knn(data.train[-3], data.test[-3], data.train$De, k=5)

CrossTable(data.test$De, predicted.out, dnn=c("Actual", "Predicted"))

34 test sample students

51 training sample students

7 students misclassified

FP 3 predicted admit, actual not admit

TN 4 predicted notadmit, actual admit

Naïve Bayes

1. Build a Naïve Bayes model with De as class variable to predict the admission class of testing sample using laplace=1. How many students have been misclassified in total? For the misclassified students, which class do they actually belong to and which class we classify them?

model <- naiveBayes(data.train[-3], data.train$De, laplace=1)

predicted.test.bayes <- predict(model, data.test[-3])

CrossTable(data.test$De, predicted.test.bayes, dnn=c("Actual", "Predicted"))

1 student was misclassified

TN 1 predicted notadmit, actual admit