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?

Ans : 85 students are included in this dataset

Code :

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

str(data)

2. 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)

Ans : 31 students belong to class admit

Code:

data$De <- ifelse(data$De =="admit",1,0); data$De <- factor(data$De)

data$De

summary(data$De)

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

Ans : Yes, sudents with higher GPA lead to higher GMAT scores

Code:

plot(x =data$GMAT, y =data$gpa, main = "graph btwn GMAT and GPA", xlab = "GMAT", ylab = "GPA")

4. 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?

Ans :

* Yes, admit class students have highest no of GPA.
* GPA is 3.403871

Code:

x <- aggregate(GPA ~ De, data=data, FUN=mean)

x

5. 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?

Ans :

* yes admit class students have highest no of GMAT
* GPA is 561.2258

Code:

Y <- aggregate(GMAT ~ De, data=data, FUN=mean)

Y

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

Ans : 36.47059 % of students belong to class admit

Code:

x <- prop.table(table(data$De))\* 100

x

7. 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.

Ans :

Original data

    GPA GMAT De

12 3.40  553  1

10 3.59  588  1

81 3.05  399  0

27 3.60  609  1

21 3.21  530  1

77 3.08  440  0

7  3.03  626  1

72 2.89  431  0

61 2.85  496  0

6  3.46  693  1

Randomized data :

   GPA GMAT De

1  2.96  596  1

2  3.14  473  1

3  3.22  482  1

4  3.29  527  1

5  3.69  505  1

6  3.46  693  1

7  3.03  626  1

8  3.19  663  1

9  3.63  447  1

10 3.59  588  1

Code:

sam <- data  
sam  
set.seed(897)  
dt.randomize <- sample(1:nrow(sam),10, replace = F)  
ad.random <- data[dt.randomize,]  
ad.random  
#original data  
data[1:10,]

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

Ans : 3.216

Code:

avg <- mean(ad.random$GPA)

avg

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

Ans :  Average of GPA is 0.5057  after normalization

Code:

nor <- function(x){

  ifelse(is.numeric(x), return ((x-min(x))/(max(x)-min(x))), return(x))

}

nor(c("GPA","GMAT"))

dt2 <- data.frame(lapply(data, nor))

summary(dt2)

10. 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?

Ans : 34 students include in testing sample

Code:

data <- dt2

nrows <- nrow(data)

nrows

train.size <- floor(nrows \* 0.6)

train.size

set.seed(1233)

sample <- sample(1:20, 10, replace = F)

sample

train.index <- sample(1:nrows, train.size,replace=F)

train.index

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

str(dt.train)

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

str(dt.test)

11. 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?

Ans : The total students misclassified is 3. They belong to class false positive,false negative. We classified them to class admit.

Code:

library(class)

predicted.test <- knn(train=dt.train[-3], test=dt.test[-3],

                       cl=dt.train$De, k=5)

install.packages('gmodels')

library(gmodels)

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

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

Ans : Mean is 0

       V1

 Min.   :-1.96876

 1st Qu.:-0.87318

 Median : 0.08255

 Mean   : 0.00000

 3rd Qu.: 0.75854

 Max.   : 1.92406

Code:

data$GPA <- scale(data$GPA)

data$GMAT <- scale(data$GMAT)

summary(data$GPA)

13. Redo steps from 10 and 11.

Code:

data <- dt2

nrows <- nrow(data)

nrows

train.size <- floor(nrows \* 0.6)

train.size

set.seed(1233)

sample <- sample(1:20, 10, replace = F)

sample

train.index <- sample(1:nrows, train.size,replace=F)

train.index

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

str(dt.train)

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

str(dt.test)

library(class)

predicted.test <- knn(train=dt.train[-3], test=dt.test[-3],

                       cl=dt.train$De, k=5)

14. Naïve Bayes

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,

Ans :

The total students misclassified is 4. They belong to class false positive, false negative. We classified them to class admit.

Code:

install.packages('e1071')

library(e1071)

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

predicted.test.naive <- predict(model, dt.test[-3])

CrossTable(dt.test[[3]], predicted.test.naive)