HW\_NB

경제학과 2020110210 공소연

2022-10-30

## Q1

library(mlbench)  
data(HouseVotes84)  
str(HouseVotes84)

## 'data.frame': 435 obs. of 17 variables:  
## $ Class: Factor w/ 2 levels "democrat","republican": 2 2 1 1 1 1 1 2 2 1 ...  
## $ V1 : Factor w/ 2 levels "n","y": 1 1 NA 1 2 1 1 1 1 2 ...  
## $ V2 : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 2 2 2 2 ...  
## $ V3 : Factor w/ 2 levels "n","y": 1 1 2 2 2 2 1 1 1 2 ...  
## $ V4 : Factor w/ 2 levels "n","y": 2 2 NA 1 1 1 2 2 2 1 ...  
## $ V5 : Factor w/ 2 levels "n","y": 2 2 2 NA 2 2 2 2 2 1 ...  
## $ V6 : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 2 2 2 1 ...  
## $ V7 : Factor w/ 2 levels "n","y": 1 1 1 1 1 1 1 1 1 2 ...  
## $ V8 : Factor w/ 2 levels "n","y": 1 1 1 1 1 1 1 1 1 2 ...  
## $ V9 : Factor w/ 2 levels "n","y": 1 1 1 1 1 1 1 1 1 2 ...  
## $ V10 : Factor w/ 2 levels "n","y": 2 1 1 1 1 1 1 1 1 1 ...  
## $ V11 : Factor w/ 2 levels "n","y": NA 1 2 2 2 1 1 1 1 1 ...  
## $ V12 : Factor w/ 2 levels "n","y": 2 2 1 1 NA 1 1 1 2 1 ...  
## $ V13 : Factor w/ 2 levels "n","y": 2 2 2 2 2 2 NA 2 2 1 ...  
## $ V14 : Factor w/ 2 levels "n","y": 2 2 2 1 2 2 2 2 2 1 ...  
## $ V15 : Factor w/ 2 levels "n","y": 1 1 1 1 2 2 2 NA 1 NA ...  
## $ V16 : Factor w/ 2 levels "n","y": 2 NA 1 2 2 2 2 2 2 NA ...

summary(HouseVotes84)

## Class V1 V2 V3 V4 V5   
## democrat :267 n :236 n :192 n :171 n :247 n :208   
## republican:168 y :187 y :195 y :253 y :177 y :212   
## NA's: 12 NA's: 48 NA's: 11 NA's: 11 NA's: 15   
## V6 V7 V8 V9 V10 V11 V12   
## n :152 n :182 n :178 n :206 n :212 n :264 n :233   
## y :272 y :239 y :242 y :207 y :216 y :150 y :171   
## NA's: 11 NA's: 14 NA's: 15 NA's: 22 NA's: 7 NA's: 21 NA's: 31   
## V13 V14 V15 V16   
## n :201 n :170 n :233 n : 62   
## y :209 y :248 y :174 y :269   
## NA's: 25 NA's: 17 NA's: 28 NA's:104

any(is.na(HouseVotes84))

## [1] TRUE

data <- na.omit(HouseVotes84)  
any(is.na(data))

## [1] FALSE

## Q2

library(caret)

## 필요한 패키지를 로딩중입니다: ggplot2

## 필요한 패키지를 로딩중입니다: lattice

parts <- createDataPartition(data$Class, p=0.8, list=F)  
training <- data[parts,]  
testing <- data[-parts,]

## Q3

library(e1071)  
data.nb <- naiveBayes(Class ~., data=training)  
data.nb

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## democrat republican   
## 0.5347594 0.4652406   
##   
## Conditional probabilities:  
## V1  
## Y n y  
## democrat 0.4100000 0.5900000  
## republican 0.8045977 0.1954023  
##   
## V2  
## Y n y  
## democrat 0.5500000 0.4500000  
## republican 0.4942529 0.5057471  
##   
## V3  
## Y n y  
## democrat 0.1200000 0.8800000  
## republican 0.8505747 0.1494253  
##   
## V4  
## Y n y  
## democrat 0.95 0.05  
## republican 0.00 1.00  
##   
## V5  
## Y n y  
## democrat 0.81000000 0.19000000  
## republican 0.02298851 0.97701149  
##   
## V6  
## Y n y  
## democrat 0.5400000 0.4600000  
## republican 0.1149425 0.8850575  
##   
## V7  
## Y n y  
## democrat 0.2200000 0.7800000  
## republican 0.7471264 0.2528736  
##   
## V8  
## Y n y  
## democrat 0.1600000 0.8400000  
## republican 0.8850575 0.1149425  
##   
## V9  
## Y n y  
## democrat 0.180000 0.820000  
## republican 0.862069 0.137931  
##   
## V10  
## Y n y  
## democrat 0.4900000 0.5100000  
## republican 0.4482759 0.5517241  
##   
## V11  
## Y n y  
## democrat 0.490000 0.510000  
## republican 0.862069 0.137931  
##   
## V12  
## Y n y  
## democrat 0.8800000 0.1200000  
## republican 0.1149425 0.8850575  
##   
## V13  
## Y n y  
## democrat 0.7000000 0.3000000  
## republican 0.1034483 0.8965517  
##   
## V14  
## Y n y  
## democrat 0.71000000 0.29000000  
## republican 0.02298851 0.97701149  
##   
## V15  
## Y n y  
## democrat 0.4000000 0.6000000  
## republican 0.8965517 0.1034483  
##   
## V16  
## Y n y  
## democrat 0.0400000 0.9600000  
## republican 0.3678161 0.6321839

options(scipen=999)  
pred.prob <- predict(data.nb,  
 newdata = testing,  
 type="raw")  
pred.class <- predict(data.nb,  
 newdata = testing,  
 type="class")  
confusionMatrix(pred.class, as.factor(testing$Class))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction democrat republican  
## democrat 19 4  
## republican 5 17  
##   
## Accuracy : 0.8   
## 95% CI : (0.654, 0.9042)  
## No Information Rate : 0.5333   
## P-Value [Acc > NIR] : 0.0001881   
##   
## Kappa : 0.5994   
##   
## Mcnemar's Test P-Value : 1.0000000   
##   
## Sensitivity : 0.7917   
## Specificity : 0.8095   
## Pos Pred Value : 0.8261   
## Neg Pred Value : 0.7727   
## Prevalence : 0.5333   
## Detection Rate : 0.4222   
## Detection Prevalence : 0.5111   
## Balanced Accuracy : 0.8006   
##   
## 'Positive' Class : democrat   
##

# Accuracy=0.933

## Q4

data.nb.l <- naiveBayes(Class ~., data=training, laplace = 1)  
data.nb.l

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## democrat republican   
## 0.5347594 0.4652406   
##   
## Conditional probabilities:  
## V1  
## Y n y  
## democrat 0.4117647 0.5882353  
## republican 0.7977528 0.2022472  
##   
## V2  
## Y n y  
## democrat 0.5490196 0.4509804  
## republican 0.4943820 0.5056180  
##   
## V3  
## Y n y  
## democrat 0.1274510 0.8725490  
## republican 0.8426966 0.1573034  
##   
## V4  
## Y n y  
## democrat 0.94117647 0.05882353  
## republican 0.01123596 0.98876404  
##   
## V5  
## Y n y  
## democrat 0.80392157 0.19607843  
## republican 0.03370787 0.96629213  
##   
## V6  
## Y n y  
## democrat 0.5392157 0.4607843  
## republican 0.1235955 0.8764045  
##   
## V7  
## Y n y  
## democrat 0.2254902 0.7745098  
## republican 0.7415730 0.2584270  
##   
## V8  
## Y n y  
## democrat 0.1666667 0.8333333  
## republican 0.8764045 0.1235955  
##   
## V9  
## Y n y  
## democrat 0.1862745 0.8137255  
## republican 0.8539326 0.1460674  
##   
## V10  
## Y n y  
## democrat 0.4901961 0.5098039  
## republican 0.4494382 0.5505618  
##   
## V11  
## Y n y  
## democrat 0.4901961 0.5098039  
## republican 0.8539326 0.1460674  
##   
## V12  
## Y n y  
## democrat 0.8725490 0.1274510  
## republican 0.1235955 0.8764045  
##   
## V13  
## Y n y  
## democrat 0.6960784 0.3039216  
## republican 0.1123596 0.8876404  
##   
## V14  
## Y n y  
## democrat 0.70588235 0.29411765  
## republican 0.03370787 0.96629213  
##   
## V15  
## Y n y  
## democrat 0.4019608 0.5980392  
## republican 0.8876404 0.1123596  
##   
## V16  
## Y n y  
## democrat 0.04901961 0.95098039  
## republican 0.37078652 0.62921348

pred.class.l <- predict(data.nb.l,  
 newdata = testing,  
 type="class")  
confusionMatrix(pred.class.l, as.factor(testing$Class))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction democrat republican  
## democrat 19 4  
## republican 5 17  
##   
## Accuracy : 0.8   
## 95% CI : (0.654, 0.9042)  
## No Information Rate : 0.5333   
## P-Value [Acc > NIR] : 0.0001881   
##   
## Kappa : 0.5994   
##   
## Mcnemar's Test P-Value : 1.0000000   
##   
## Sensitivity : 0.7917   
## Specificity : 0.8095   
## Pos Pred Value : 0.8261   
## Neg Pred Value : 0.7727   
## Prevalence : 0.5333   
## Detection Rate : 0.4222   
## Detection Prevalence : 0.5111   
## Balanced Accuracy : 0.8006   
##   
## 'Positive' Class : democrat   
##

# Accuracy=0.933