```
#라이브러리 정리
library(gridExtra)
library(pastecs)
library(ggplot2)
library(gmodels)
source('C:/Users/Kwon/Desktop/Downloads/자료폴더_2311109/EDA/descriptive_analytics_utils.R')
source("C:/Users/Kwon/Desktop/Downloads/자료폴더_2311109/ModelComparison/performance_plot_utils.R")  # plotting metric results
library(corrplot)
library(dplyr)
library(class)
library(DMwR2)
library(ggcorrplot)
library(e1071)
library(caret) # model training and evaluation
library(ROCR) # model evaluation
library(rpart)
library(rpart.plot)
library(nnet)
library(car)
#=====================================
Sys.getlocale()
Sys.setlocale("LC_ALL", "C")
Sys.setlocale("LC_ALL", "Korean")
Sys.setlocale("LC_ALL", "Korean") #일부러 2번작성
#데이터 불러오기 오류 해결 코드

data=read.csv("C:/Users/Kwon/Desktop/Downloads/2021_성인독서_데이터.csv", header = T)
head(data)
```

```
str(data)


#================================================================
# 1. 데이터 전처리=================================================
# 1.1 연속형변수=================================================
# 1.1.1 여가시간(freetime_total)  // 분단위으로 통합
weekday.freetime.total= data$weekday.freetime.h.*60 +data$weekday.freetime.m.
weekend.freetime.total= data$weekend.freetime.h.*60 +data$weekend.freetime.m.
data$freetime_total=weekday.freetime.total+weekend.freetime.total
str(data$freetime_total)

# 1.1.2 종이책 독서시간(book_total)  // 분단위으로 통합
book_weekday.total=data$book_weekday.h.*60 +data$book_weekday.m.
book_weekend.total=data$book_weekend.h.*60 +data$book_weekend.m.
book_total=book_weekday.total+book_weekend.total
sum(is.na(book_total))
str(book_total)

# 1.1.3 전자책 독서시간(ebook_total)  // 분단위로 통합
ebook_weekday.total=data$ebook_weekday.h.*60 +data$ebook_weekday.m.
ebook_weekend.total=data$ebook_weekend.h.*60 +data$ebook_weekend.m.
ebook_total=ebook_weekday.total+ebook_weekend.total
sum(is.na(ebook_total))
str(ebook_total)

# 1.1.4 독서 선호도(read.preference)
sum(is.na(data$read.preference))
data$read.preference=abs(data$read.preference-6)
str(data$read.preference)
```

```
# 1.1.5 연령(age)

sum(is.na(data$age))


# 1.1.6 가구소득(household.income)

sum(is.na(data$household.income))




# 1.2 범주형 변수==================================================

# 1.2.1 종이책보다 전자책을 더 선호하는 경우(ebookmore)

data$ebookmore=ifelse(ebook_total-book_total>0,1,0)

str(data$ebookmore)

data$ebookmore=as.factor(data$ebookmore)

str(data$ebookmore)


# 1.2.2 빈도(read.frequency) // 숫자가 클수록 빈번하도록 수정

sum(is.na(data$read.frequency))

data$read.frequency=abs(data$read.frequency-6)

str(data$read.frequency)

data$read.frequency=as.factor(data$read.frequency)

str(data$read.frequency)


# 1.2.3 성별(gender)

# 1: 남 | 2: 여

sum(is.na(data$gender))

str(data$gender)

data$gender=as.factor(data$gender)

str(data$gender)
```

```
# 1.2.4 독서장소(read.place)

# 1: 집 | 2: 직장(학교) | 3: 이동시 | 4: 도서관

# 5: 서점 | 6: 카페 | 7: 어디서든 | 8: 기타

sum(is.na(data$read.place))

str(data$read.place)

data$read.place = as.factor(data$read.place )

str(data$read.place)


# 1.2.5 도서선택 이용정보(choice_information)

# 1: 책 직접보고  | 2: 신문/잡지 소개 | 3: 텔레비전/라디오 소개 | 4: 인터넷 소개

# 5: 지인 추천  |  6: 유명인 추천  |  7: 베스트셀러 목록  | 8: 추천도서 목록

# 9: 드라마/영화 원작   | 10: SNS 소개  |  11: 유튜브 소개  | 12: 기타

sum(is.na(data$choice_information))

str(data$choice_information)

data$choice_information=as.factor(data$choice_information)

str(data$choice_information)


# 1.2.6 직업(job)

# 1:관리자 | 2:전문가 | 3:사무종사자 | 4:서비스종사자 | 5:판매종사자

# 6:농림어업 종사자 | 7:기능종사자 | 8:장치종사자 | 9:단순 노무 종사자 | 10:군인

# 11:자영업종사자 | 12:학생 | 13:전업주부 | 14:무직  15: 기타

sum(is.na(data$job))

str(data$job)

data$job=as.factor(data$job)

str(data$job)



# 1.2.7 도서관 이용 유무 (library)
```

```
# 1: 이용한 적이 있다 | 0: 이용한 적이 없다

sum(is.na(data$library))

str(data$library)

data$library = factor(data$library, levels=c(1,2), labels=c(1,0))

data$library

str(data$library)


# 1.2.8 지역(area)

# 1: 대도시 | 2: 중소도시 | 3: 읍면

sum(is.na(data$area))

str(data$area)

data$area=as.factor(data$area)

str(data$area)

data$area


# 1.2.9 독서활동 여부(reading.activity)

# 1: 참여한 적이 있다 | 0: 참여한 적이 없다

sum(is.na(data$reading.activity))

str(data$reading.activity)

data$reading.activity = factor(data$reading.activity, levels=c(1,2), labels=c(1,0))

data$reading.activity

str(data$reading.activity)


# 1.2.10 독서동아리 참여여부(reading.club)

# 1: 참여한 적이 있다 | 0: 참여한 적이 없다

sum(is.na(data$reading.club))

str(data$reading.club)

data$reading.club <- factor(data$reading.club, levels=c(1,2), labels=c(1,0))

data$reading.club
```

```
str(data$reading.club)


# 1.2.11 최종 학력(최종.학력)
# 1: 교육을 안 받았음 | 2: 초등학교 | 3: 중학교 | 4: 고등학교
# 5: 대학(4년제 미만) | 6: 대학(4년제 이상) | 7: 대학원 석사과정 | 8: 대학원 박사과정
sum(is.na(data$최종.학력))
str(data$최종.학력)
data$최종.학력 <- factor(data$최종.학력)
str(data$최종.학력)
data$최종.학력



# 1.3 데이터 세트 정리 (필요 변수만 남김)==========================
head(data)
data=data[,-c(1,2,3,4,18,19,20,21,22,23,24,25)]
head(data)
rownames(data)<- NULL
str(data)
#=================================================================



#=================================================================
# 2. 분포확인=====================================================
# 2.1 연속형 변수의 상자그림과 히스토그램=========================
# 2.1.1 여가시간(freetime_total)
```

```
boxplot(data$freetime_total, main="Boxplot of freetime", ylab="freetime")

hist(data$freetime_total, main="Histogram of freetime", xlab="freetime")


# 2.1.2 독서 선호도(read.preference)

boxplot(data$read.preference, main="Boxplot of read.preference", ylab="read.preference")

hist(data$read.preference, main="Histogram of read.preference", xlab="read.preference")


# 2.1.3 연령(age)

boxplot(data$age, main="Boxplot of age", ylab="age")

hist(data$age, main="Histogram of age", xlab="age")


# 2.1.4 가구소득(household.income)

boxplot(data$household.income, main="Boxplot of household.income", ylab="household.income")

hist(data$household.income, main="Histogram of household.income", xlab="household.income")



# 2.2 범주형 변수의 빈도표=====================================

# 2.2.1 전자책을 더 선호하는 경우(ebookmore)

ebookmore_table <- table(data$ebookmore)

print(ebookmore_table)

barplot(ebookmore_table, main="Frequency of ebookmore", xlab="ebookmore", ylab="Frequency")


# 2.2.2 독서빈도(read.frequency)

read.frequency_table <- table(data$read.frequency)

print(read.frequency_table)

barplot(read.frequency_table,main="Frequency of read frequency", xlab="read.frequency",ylab="Frequency")


# 2.2.2 성별(gender)

gender_table <- table(data$gender)
```

```
print(gender_table)

barplot(gender_table,main="Frequency of gender", xlab="gender",ylab="Frequency")


# 2.2.3 독서장소(read.place)

read.place_table <- table(data$read.place)

print(read.place_table)

barplot(read.place_table, main="Frequency of read place", xlab="read.place", ylab="Frequency")


# 2.2.4 도서선택 이용정보(choice_information)

choice_information_table <- table(data$choice_information)

print(choice_information_table)

barplot(choice_information_table, main="Frequency of choice information", xlab="choice_information", ylab="Frequency")


# 2.2.5 직업(job)

job_table <- table(data$job)

print(job_table)

barplot(job_table, main="Frequency of job", xlab="job",ylab="Frequency")

#1:관리자 | 2:전문가| 10:군인 | 15: 기타 를 모두 15:기타 변경.

data$job= recode(data$job,'1=15;2=15;3=3;4=4;5=5;6=6;7=7;8=8;9=9;10=15;11=11;12=12;13=13;14=14;15=15')

data$job=as.factor(data$job)

str(data$job)

job_table <- table(data$job)

print(job_table)

barplot(job_table, main="Frequency of job New", xlab="job",ylab="Frequency")


# 2.2.6 도서관 이용 유무(library)

library_table <- table(data$library)
```

```
print(library_table)

barplot(library_table, main="Frequency of library", xlab="Library", ylab="Frequency")


# 2.2.7 지역(area)

area_table <- table(data$area)

print(area_table)

barplot(area_table, main="Frequency of area", xlab="area", ylab="Frequency")


# 2.2.8 독서활동 여부(reading.activity)

reading.activity_table <- table(data$reading.activity)

print(reading.activity_table)

barplot(reading.activity_table, main="Frequency of reading activity", xlab="reading.activity", ylab="Frequency")


# 2.2.9 독서동아리 참여여부(reading.club)

reading.club_table <- table(data$reading.club)

print(reading.club_table)

barplot(reading.club_table, main="Frequency of reading club", xlab="reading.club", ylab="Frequency")


# 2.2.10 최종 학력(최종.학력)

최종.학력_table <- table(data$최종.학력)

print(최종.학력_table)

barplot(최종.학력_table, main="Frequency of 최종 학력", xlab="area", ylab="Frequency")


#=================================================================


#=================================================================

# 3. KNN (결측치 전환)============================================

# 3.1 결측값 행이 제거된 데이터=====================================

data2=data%>% filter(!is.na(data$read.place))
```

```
tail(data2)


# 3.2 가장 적합한 k값 찾기=========================================

# 3.2.1 독서장소(read.place)

set.seed(0)

ind <- sample(x=2, nrow(data2), replace=T, prob=c(0.7,0.3))

train <- data2[ind==1,]

test <- data2[ind==2,]

dim(train)

dim(test)

train_labels = train[, 8]

test_labels = test[, 8]



k1=c(5,10,15,20,25)

rr1=rep(0,length(k1))


for (i in 1:length(k1)) {

  test_pred = knn(train = train, test = test,   cl = train_labels, k = k1[i])

  #예측결과test_pred

  #CrossTable(x=test_labels,y=test_pred, prop.chisq = F)

  tab=table(test_labels,test_pred)

  rr1[i]=sum(tab[row(tab) == col(tab)])/sum(tab) #정분류율

}

cbind(k1,rr1)



k2=c(5:15)

rr2=rep(0,length(k2))
```

```
for (i in 1:length(k2)) {
  test_pred = knn(train = train, test = test,   cl = train_labels, k = k2[i])
  #예측결과test_pred
  #CrossTable(x=test_labels,y=test_pred, prop.chisq = F)
  tab=table(test_labels,test_pred)
  rr2[i]=sum(tab[row(tab) == col(tab)])/sum(tab) #정분류율
}
cbind(k2,rr2)



# 3.2.2 도서선택 이용정보(choice_information)
train_labels = train[, 9] # train 데이터의 특정변수의 레이블을 따로 저장
test_labels = test[, 9]  # test 데이터의 특정변수의 레이블을 따로 저장

k3=c(5,10,15,20,25)
rr3=rep(0,length(k3))

for (i in 1:length(k3)) {
  test_pred = knn(train = train, test = test,   cl = train_labels, k = k3[i])
  #예측결과test_pred
  #CrossTable(x=test_labels,y=test_pred, prop.chisq = F)
  tab=table(test_labels,test_pred)
  rr3[i]=sum(tab[row(tab) == col(tab)])/sum(tab) #정분류율
}
cbind(k3,rr3)

k4=c(5:15)
rr4=rep(0,length(k4))
```

```
for (i in 1:length(k4)) {
  test_pred = knn(train = train, test = test,   cl = train_labels, k = k4[i])
  #예측결과test_pred
  #CrossTable(x=test_labels,y=test_pred, prop.chisq = F)
  tab=table(test_labels,test_pred)
  rr4[i]=sum(tab[row(tab) == col(tab)])/sum(tab) #정분류율
}
cbind(k4,rr4)
cbind(k4,rr2,rr4)

#k=8 일때 두 변수 모두에서 정분류율이 높아보인다.
#>>>>> 한 변수에 결측값이 있는경우에만 적용되고, 두 변수에 결측값이 있을때는 안된다


# 3.3 KNN 결측치 채우기==========================================
#data3<- knnImputation(data,k=8) #
#https://blog.naver.com/pmw9440/222075083203
#>>>>>인터넷에서 찾은 KNN, 앞에서 가장 효과가 좋았던 8로.
#할때마다 데이터가 달라질 수 있기 때문에 임의로 데이터 저장

#######
#저장해둔 데이터 사용할 때
data3=read.csv('D:/대학교/성인_독서_데이터/datafinal.csv', header = T)
str(data3)
categorical.vars=c('gender' ,'area','최종.학력', 'job', 'library',
                   'read.place','choice_information','reading.activity' ,
                   'reading.club','read.frequency','ebookmore')
numerical.vars=c('freetime_total', 'age', 'read.preference' ,'household.income')
```

```
data3 <- to.factors(df = data3, variables = categorical.vars)

str(data3)

#######


head(data)

head(data3)




# 3.4.1 결측치 채운 후 독서장소(read.place)

read.place_table2 <- table(data3$read.place)

print(read.place_table2)

barplot(read.place_table2, main="Frequency of read place New", xlab="read.place", ylab="Frequency")


# 3.4.2 결측치 채운 후 도서선택 이용정보(choice_information)

choice_information_table2 <- table(data3$choice_information)

print(choice_information_table2)

barplot(choice_information_table2, main="Frequency of choice information New", xlab="choice_information", ylab="Frequency")

#=================================================================




#=================================================================

# 4. 독립성 검정==================================================

# 4.1 연속형변수==================================================

attach(data3)

str(data3)
```

```
# 4.1.1 여가시간(freetime_total)

freetime_total1 <- data3[ebookmore==1,"freetime_total"]

freetime_total0 <- data3[ebookmore==0,"freetime_total"]

t.test(freetime_total1,freetime_total0)


# 4.1.2 독서선호도(read.preference)

read.preference1 <- data3[ebookmore==1,"read.preference"]

read.preference0 <- data3[ebookmore==0,"read.preference"]

t.test(read.preference1,read.preference0)


# 4.1.3 연령(age)

age1 <- data3[ebookmore==1,"age"]

age0 <- data3[ebookmore==0,"age"]

t.test(age1,age0)


# 4.1.4 가구소득(household.income)

household.income1 <- data3[ebookmore==1,"household.income"]

household.income0 <- data3[ebookmore==0,"household.income"]

t.test(household.income1,household.income0)
#귀무가설: 두 집단의 평균이 같다

#ttest결과 p값이 0.05보다 작아서 귀무가설을 기각

#두 집단의 평균이 다르다고 할 수 있다

#따라서 반응변수와 설명변수들 간의 관계가 있다고 할 수 있다.



# 4.2 범주형 변수==================================================

# 4.2.1 지역(area)

get.contingency.table(ebookmore, area, stat.tests = T)
```

4.2.2 최종학력(최종.학력)

get.contingency.table(ebookmore,최종.학력, stat.tests = T)

4.2.3 직업(job)

get.contingency.table(ebookmore,job, stat.tests = T)

4.2.4 독서장소(read.place)

get.contingency.table(ebookmore,read.place, stat.tests = T)

4.2.5 도서선택이용정보(choice_information)

get.contingency.table(ebookmore,choice_information, stat.tests = T)

4.2.6 독서빈도(read.frequency)

get.contingency.table(ebookmore,read.frequency, stat.tests = T)

4.3 이항변수==========

4.3.1 성별(gender)

fisher.test(ebookmore, gender)

chisq.test(ebookmore, gender)

4.3.2 도서관이용유무(library)

fisher.test(ebookmore, library)

chisq.test(ebookmore, library)

4.3.3 독서활동여부(reading.activity)

fisher.test(ebookmore, reading.activity)

chisq.test(ebookmore, reading.activity)

```
# 4.3.4 독서동아리 참여여부(reading.club)

fisher.test(ebookmore, reading.club)

chisq.test(ebookmore, reading.club)


#반응변수 : ebookmore

#모든 설명변수에 대해서 p값<0.05

#귀무가설: 두 범주형 변수는 서로독립이다(관계가 없다)

#대립가설: 두 범주형 변수는 서로독립이 아니다(관계가 있다)

#모든 설명변수들이 귀무가설을 기각하기 때문에

#모든 변수들은 반응변수인 ebookmore과 관계가 있다

#=====================================================================

categorical.vars=c('gender' ,'area','최종.학력', 'job', 'library',

                   'read.place','choice_information','reading.activity' ,

                   'reading.club','read.frequency','ebookmore')

numerical.vars=c('freetime_total', 'age', 'read.preference' ,'household.income')


nume=data3[numerical.vars]

nume

str(nume)

#연속형 변수간에 상관관계가 있는지 확인.

ggcorrplot(cor(nume), lab = T, hc.order = T, type = "lower", outline.color = "white")

cor(nume) #상관계수 확인결과 관계가 있어보이는 변수는 없다.


#=====================================================================


#=====================================================================

# 5. 머신러닝 모델링 ================================================

# 5.1 train set/ test set (60:40)=============================================

#데이터 분할
```

```
#set.seed(10)

#indexes<-sample(1:nrow(data3), size = 0.6*nrow(data3))

#원래는 직접 분류를 해야되지만 매번 값이 바뀌기 때문에 따로 저장한 인덱스 사용

#write.csv(indexes,file="indexes.csv") #저장할 때 사용한 코드

#######

indexes=read.csv("D:/대학교/성인_독서_데이터/indexes.csv", header = F)

indexes=indexes$V1

indexes

#######


train.data3<-data3[indexes,]

head(train.data3)

test.data3<-data3[-indexes,]

test.feature.vars<-test.data3[,-15]

head(test.feature.vars)

test.class.var<-test.data3[,15]

head(test.class.var)


# 5.1 로지스틱모형==================================================

# build a logistic regression model

formula.init <- "ebookmore ~ ."

formula.init <- as.formula(formula.init)

lr.model <- glm(formula=formula.init, data=train.data3, family="binomial")


# view model details

summary(lr.model)


# perform and evaluate predictions

lr.predictions <- predict(lr.model, test.data3, type="response")
```

```
lr.predictions <- round(lr.predictions)

confusionMatrix(data=as.factor(lr.predictions), reference=as.factor(test.class.var), positive='1')



## glm specific feature selection

formula <- "ebookmore ~ ."

formula <- as.formula(formula)

control <- trainControl(method="repeatedcv", number=10, repeats=2)

model <- train(formula, data=train.data3, method="glm", trControl=control)

#오류나는데 아래코드는 작동

importance #원래 없던 변수

importance <- varImp(model, scale=FALSE)

plot(importance) #변수 잘 작동



# build new model with selected features

formula.new <- formula.new <- "ebookmore ~ age +choice_information + read.place +  household.income +freetime_total+gender"

formula.new <- as.formula(formula.new)

lr.model.new <- glm(formula=formula.new, data=train.data3, family="binomial")



# view model details

summary(lr.model.new)



# perform and evaluate predictions

#lr.model.new 축소모형

lr.predictions.new <- predict(lr.model.new, test.data3, type="response")
```

```
lr.predictions.new <- round(lr.predictions.new)

confusionMatrix(data=as.factor(lr.predictions.new), reference=as.factor(test.class.var), positive='1')



## model performance evaluations


# plot best model evaluation metric curves

#전체모형

lr.model.best <- lr.model

lr.prediction.values <- predict(lr.model.best, test.feature.vars, type="response")


predictions <- prediction(lr.prediction.values, test.class.var)

par(mfrow=c(1,2))

plot.roc.curve(predictions, title.text="LR ROC Curve")

plot.pr.curve(predictions, title.text="LR Precision/Recall Curve")


#축소모형

lr.model.best2 <- lr.model.new

lr.prediction.values2 <- predict(lr.model.best2, test.feature.vars, type="response")

predictions2 <- prediction(lr.prediction.values2, test.class.var)

par(mfrow=c(1,2))

plot.roc.curve(predictions2, title.text="LR ROC Curve")

plot.pr.curve(predictions2, title.text="LR Precision/Recall Curve")


# 5.2 svm=====================================================

#데이터 분할 60:40

#set.seed(10)

#indexes<-sample(1:nrow(data3), size = 0.6*nrow(data3))

#train.data3<-data3[indexes,]
```

```
#head(train.data3)

#test.data3<-data3[-indexes,]

#test.feature.vars<-test.data3[,-15]

#head(test.feature.vars)

#test.class.var<-test.data3[,15]

#head(test.class.var)

#앞의 데이터셋 그대로 사용하기 때문에 생략가능



## build initial model with training data

formula.init.data3<- "ebookmore ~ ."

formula.init.data3 <- as.formula(formula.init.data3)


svm.model.data3 <- svm(formula=formula.init.data3, data=train.data3,

                       kernel="radial", cost=100, gamma=1)


summary(svm.model.data3)



svm.predictions.data3 <- predict(svm.model.data3, test.feature.vars)

confusionMatrix(data=svm.predictions.data3, reference=test.class.var, positive="1")



##########


## svm specific feature selection

formula.init.data3 <- "ebookmore ~ ."

formula.init.data3 <- as.formula(formula.init.data3)

control.data3 <- trainControl(method="repeatedcv", number=10, repeats=2)
```

```
model.data3 <- train(formula.init.data3, data=train.data3, method="svmRadial",
                     trControl=control.data3)
#오류뜨는데 작동
importance <- varImp(model.data3, scale=FALSE)
plot(importance, cex.lab=0.5)



#축소모형
formula.new <- "ebookmore ~  read.frequency + age + 최종.학력 + read.preference + choice_information +read.place"
formula.new <- as.formula(formula.new)
svm.model.new <- svm(formula=formula.new, data=train.data3,
                     kernel="radial", cost=10, gamma=0.25)

## predict results with new model on test data
svm.predictions.new <- predict(svm.model.new, test.feature.vars)

## new model performance evaluation
confusionMatrix(data=svm.predictions.new, reference=test.class.var, positive="1")



## hyperparameter optimizations

# run grid search
cost.weights <- c(0.1, 10, 100)
gamma.weights <- c(0.01, 0.25, 0.5, 1)
tuning.results <- tune(svm, formula.new,
                       data = train.data3, #kernel="Radial",
                       ranges=list(cost=cost.weights, gamma=gamma.weights))
```

```
# view optimization results

print(tuning.results)




# plot results

plot(tuning.results, cex.main=0.6, cex.lab=0.8,xaxs="i", yaxs="i")


# get best model and evaluate predictions

svm.model.best = tuning.results$best.model

svm.predictions.best <- predict(svm.model.best, test.feature.vars)

confusionMatrix(data=svm.predictions.best, reference=test.class.var, positive="1")




# plot best model evaluation metric curves

#전체모형

svm.predictions.best <- predict(svm.model.data3 , test.feature.vars, decision.values = T)

svm.prediction.values <- attributes(svm.predictions.best)$decision.values


predictions <- prediction(svm.prediction.values, test.class.var)

par(mfrow=c(1,2))

plot.roc.curve(predictions, title.text="SVM ROC Curve")

plot.pr.curve(predictions, title.text="SVM Precision/Recall Curve")


#축소모형

predictions.new <- predict(svm.model.new, test.feature.vars, decision.values = T)

svm.prediction.values.new <- attributes(predictions.new)$decision.values
```

```
predictions.new <- prediction(svm.prediction.values.new, test.class.var)

par(mfrow=c(1,2))

plot.roc.curve(predictions.new, title.text="SVM ROC Curve")

plot.pr.curve(predictions.new, title.text="SVM Precision/Recall Curve")


# 5.3 의사결정나무============================================================

## 초기 모델링 훈련 코드

formula.init <- "ebookmore ~ ."

formula.init <- as.formula(formula.init)

dt.model <- rpart(formula=formula.init, method="class",data=train.data3, control = rpart.control(minsplit=20, cp=0.05))


## 데이터 예측 수행 및 결과 평가

dt.predictions <- predict(dt.model, test.feature.vars, type="class")

confusionMatrix(data=dt.predictions, reference=test.class.var, positive="1")



## 의사결정나무 특성 선택

formula.init <- "ebookmore ~ ."

formula.init <- as.formula(formula.init)

control <- trainControl(method="repeatedcv", number=10, repeats=2)

model <- train(formula.init, data=train.data3, method="rpart", trControl=control)

importance <- varImp(model, scale=FALSE)

plot(importance, cex.lab=0.5)


## 특성 선택을 기반으로 새로운 의사결정나무 모델 구축

formula.new <- "ebookmore ~ read.frequency+ age+ read.preference +  최종.학력  + library +choice_information"

formula.new <- as.formula(formula.new)

dt.model.new  <-  rpart(formula=formula.new,  method="class",data=train.data3,    control  =  rpart.control(minsplit=20, cp=0.05), parms = list(prior = c(0.6, 0.4)))
```

```
dt.predictions.new <- predict(dt.model.new, test.feature.vars, type="class")

confusionMatrix(data=dt.predictions.new, reference=test.class.var, positive="1")


## 의사결정나무 시각화

dt.model.best <- dt.model.new

print(dt.model.best)


par(mfrow=c(1,1))

prp(dt.model.best, type=1, extra=3, varlen=0, faclen=0)


## 모델의 평가 지표 곡선 시각화

#전체모형

dt.predictions.full <- predict(dt.model, test.feature.vars, type="prob")

dt.prediction.values <- dt.predictions.full[,2]

predictions <- prediction(dt.prediction.values, test.data3[,15])

par(mfrow=c(1,2))

plot.roc.curve(predictions, title.text="DT ROC Curve")

plot.pr.curve(predictions, title.text="DT Precision/Recall Curve")



#축소모형

dt.predictions.best <- predict(dt.model.best, test.feature.vars, type="prob")

dt.prediction.values2 <- dt.predictions.best[,2]

predictions2 <- prediction(dt.prediction.values2, test.data3[,15])

par(mfrow=c(1,2))

plot.roc.curve(predictions2, title.text="DT ROC Curve")

plot.pr.curve(predictions2, title.text="DT Precision/Recall Curve")


# 5.4 인공신경망=====================================================
```

```
formula.init <- "ebookmore ~ ."

formula.init <- as.formula(formula.init)

nn.model <- train(formula.init, data = train.data3, method="nnet")

nn.predictions <- predict(nn.model, test.feature.vars, type="raw")

confusionMatrix(data=nn.predictions, reference=test.class.var, positive="1")



# nn feature selection (신경망 특성 선택)

formula.init <- "ebookmore ~ ."

formula.init <- as.formula(formula.init)

control <- trainControl(method="repeatedcv", number=10, repeats=2)

nn_model <- train(formula.init, data=train.data3, method="nnet",

                  trControl=control)

importance <- varImp(nn_model, scale=FALSE)

plot(importance, cex.lab=0.5)


# Reduced model by importance

formula.init <- "ebookmore ~최종.학력+area+ read.frequency +job+ choice_information+ read.place"

formula.init <- as.formula(formula.init)

nn.model2 <- train(formula.init, data = train.data3, method="nnet")

nn.predictions2 <- predict(nn.model2, test.feature.vars)


confusionMatrix(data=nn.predictions2, reference=test.class.var, positive="1")



## plot model evaluation metric curves

nn.predictions <- predict(nn.model, test.feature.vars, type="prob")

nn.prediction.values <- nn.predictions[,2]

predictions <- prediction(nn.prediction.values, test.class.var)
```

```
nn.predictions2 <- predict(nn.model2, test.feature.vars, type="prob")

nn.prediction2.values <- nn.predictions2[,2]

predictions2 <- prediction(nn.prediction2.values, test.class.var)

par(mfrow=c(1,2))

plot.roc.curve(predictions, title.text="nnet ROC Curve")

plot.roc.curve(predictions2, title.text="nnet ROC Curve")
```