

3. SVM SN

```
rm(list=ls())
graphics.off()
setwd("~/Tobigs/homework_0801/SVM")
library(e1071)
library(kernlab)
library(caret)
SN=read.csv('SN_ad.csv')
str(SN)
#ID 제거
SN=SN[-1]
SN$Purchased=as.factor(SN$Purchased)
#남성 여성 분리
SN.male=subset(SN,Gender=='Male')
SN.female=subset(SN,Gender=='Female')
summary(SN$Age)
summary(SN$EstimatedSalary)
#구매함 = blue, 구매안함 = red
#어떤 물건인지는 몰라도 Gender 에 대해서는 구매 여부에 상관이 없음
#영향을 주는 변수는 Age 와 Salary
while(1){
  par(mfrow=c(2,2))
  plot(SN$Age,SN$Gender,pch=c(4,19)[SN$Purchased],
       col=c('red','blue')[SN$Purchased],xlim=c(18,60),ylim=c(1,2),xlab='Age',ylab='Salary
(*1000)',main='원본 Gender')
  plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN$Purchased],

col=c('red','blue')[SN$Purchased],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='원본')
  plot(SN.male$Age,SN.male$EstimatedSalary/1000,pch=c(4,19)[SN.male$Purchased],

col=c('red','blue')[SN.male$Purchased],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='Male')
  plot(SN.female$Age,SN.female$EstimatedSalary/1000,pch=c(4,19)[SN.female$Purchased],

col=c('red','blue')[SN.female$Purchased],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='Female')
  break
}
#svm
svm.model<- svm(Purchased~., data=SN, gamma=1, cost=1)
addmargins(table(SN$Purchased, svm.model$fitted))
while(1){
  par(mfrow=c(2,2))
  plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN$Purchased],

col=c('red','blue')[SN$Purchased],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='원본')
  plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[svm.model$fitted],

col=c('red','blue')[svm.model$fitted],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='g=1 c=1')
  break
}
sum(SN$Purchased==svm.model$fitted)/length(SN$Purchased) #0.9175
#polynomial 커널
SN.model1 <- ksvm(Purchased ~ ., data = SN,
                  type = "C-bsvc", kernel = "polydot",
                  kpar = list(degree = 4), C = 10,
                  prob.model = TRUE)
SN.poly=predict(SN.model1, subset(SN,select=-c(Purchased)))
while(1){
```

```

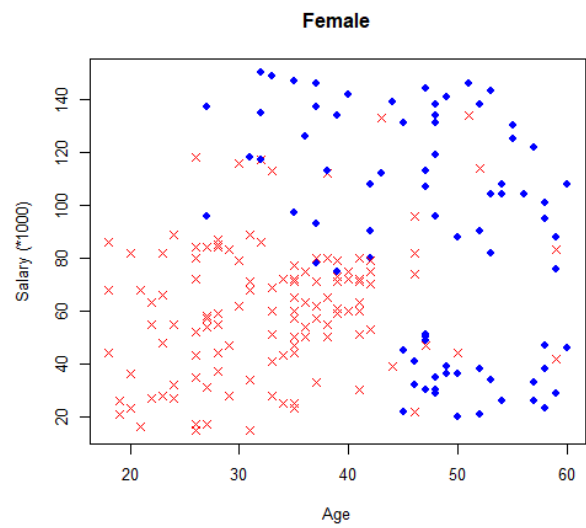
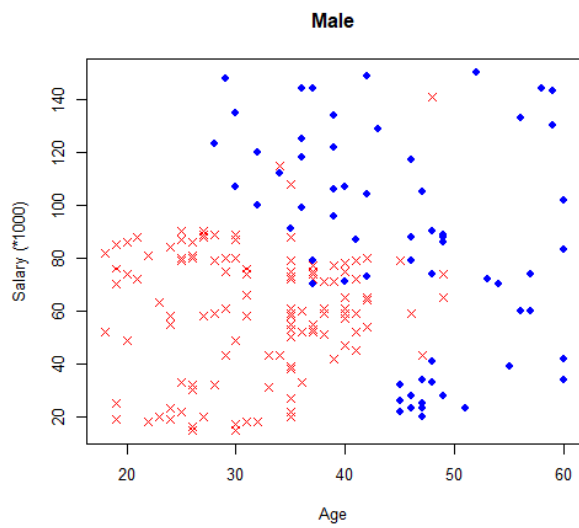
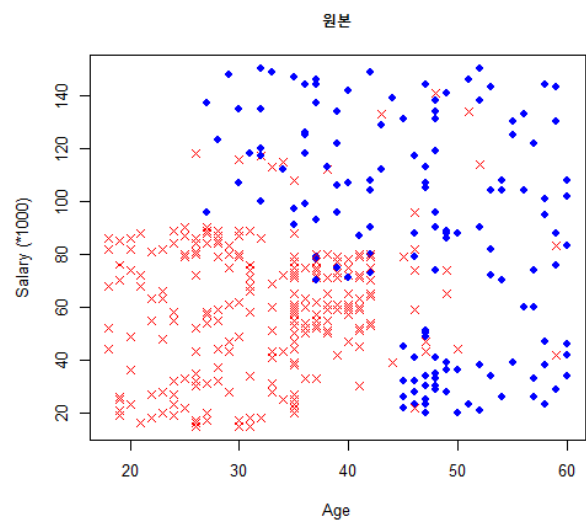
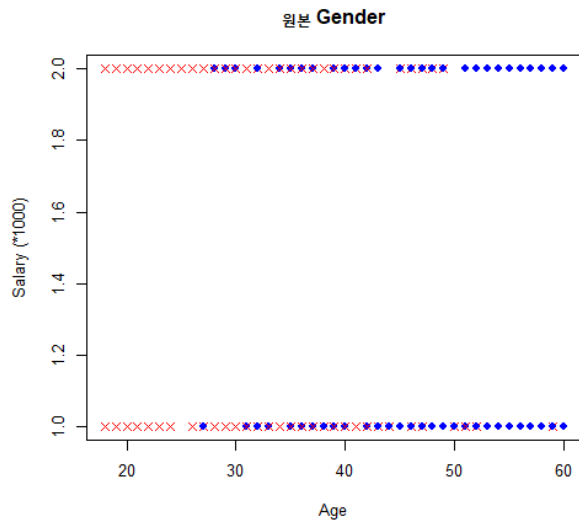
par(mfrow=c(2,2))
plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN$Purchased],

col=c('red','blue')[SN$Purchased],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='원본')
plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN.poly],
col=c('red','blue')[SN.poly],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='poly 4th order g=10')
break
}
sum(SN$Purchased==SN.poly)/length(SN$Purchased) #0.9175
#rbf 커널
SN.model2 <- ksvm(Purchased ~ ., data = SN,
type = "C-bsvc", kernel = "rbfdot",
kpar = list(sigma = 100), C = 10,
prob.model = TRUE)
SN.rbf=predict(SN.model2, subset(SN,select=-c(Purchased)))
while(1){
par(mfrow=c(2,2))
plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN$Purchased],

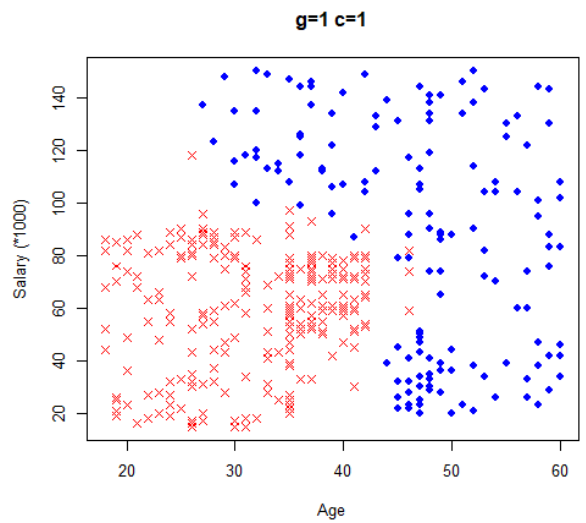
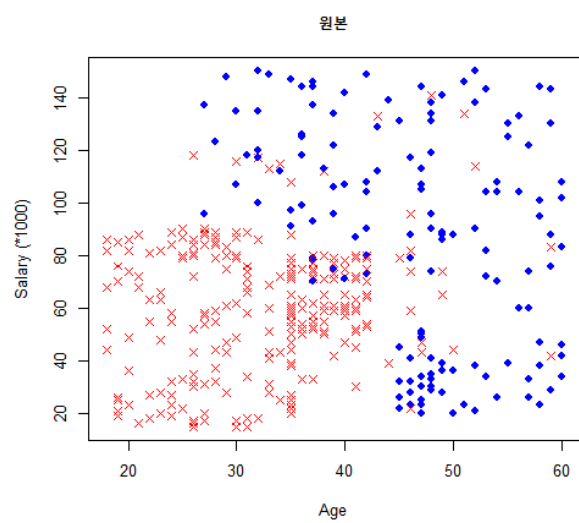
col=c('red','blue')[SN$Purchased],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='원본')
plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN.rbf],
col=c('red','blue')[SN.rbf],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='rbf g=100 C=10')
break
}
sum(SN$Purchased==SN.rbf)/length(SN$Purchased) #0.9975
#Hyperbolic 커널
SN.model3 <- ksvm(Purchased ~ ., data = SN,
type = "C-bsvc", kernel = "tanhdot",
kpar = list(scale = 0.005), C = 100,
prob.model = TRUE)
SN.tanh=predict(SN.model3, subset(SN,select=-c(Purchased)))
while(1){
par(mfrow=c(2,2))
plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN$Purchased],

col=c('red','blue')[SN$Purchased],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='원본')
plot(SN$Age,SN$EstimatedSalary/1000,pch=c(4,19)[SN.tanh],
col=c('red','blue')[SN.tanh],xlim=c(18,60),ylim=c(15,150),xlab='Age',ylab='Salary
(*1000)',main='tanh scale=0.005 C=100')
break
}
sum(SN$Purchased==SN.tanh)/length(SN$Purchased) #0.83
#rbf 모델이 0.9975 로 가장 높았다.
#rbf > poly 4th > tanh

```

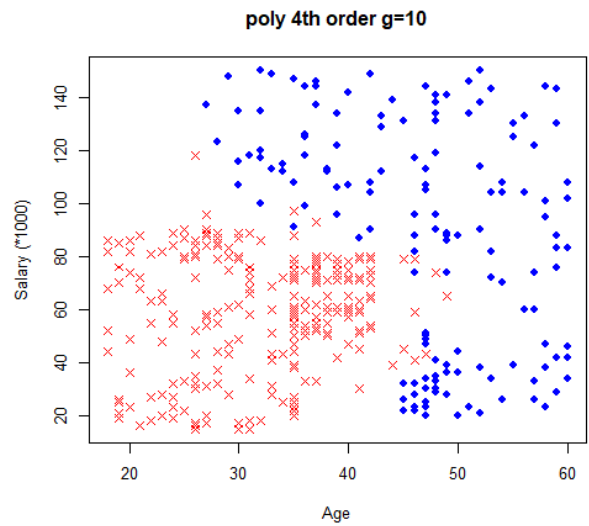
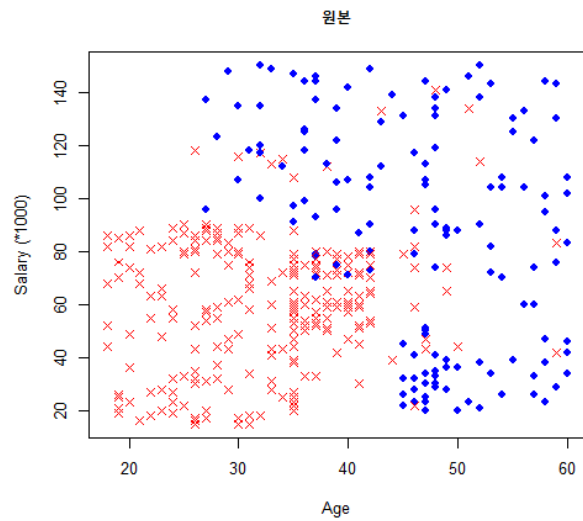


원본 데이터 플롯 (파란점=구매, 빨간 X=구매 X)

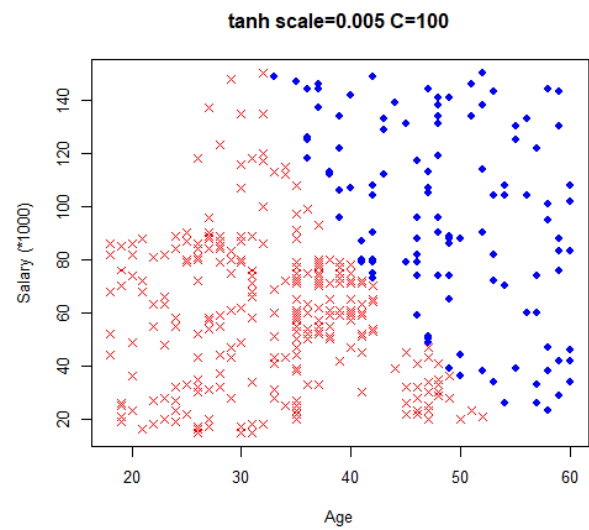
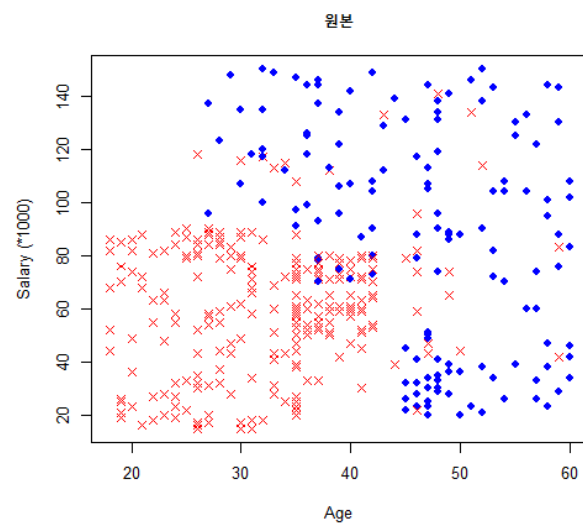


원본과 SVM 결과 비교 플롯

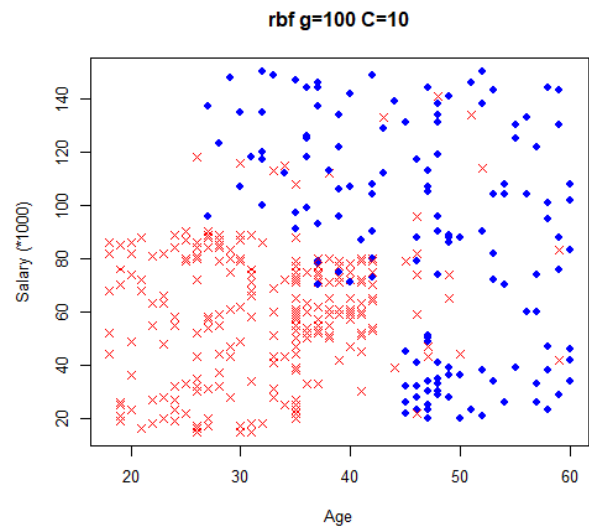
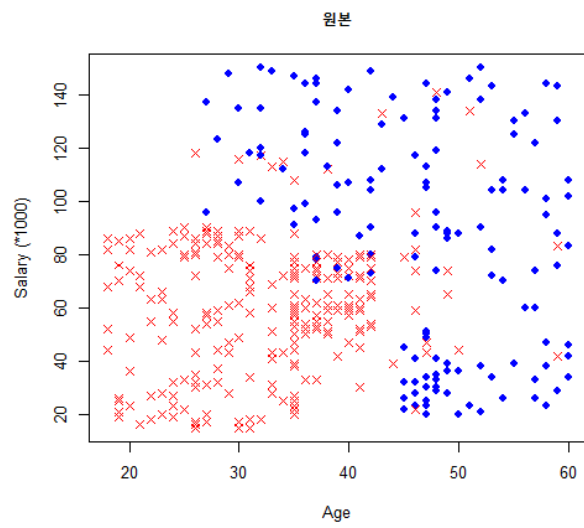
정확도 0.9175



원본과 Polynomial 결과 비교 플롯
정확도 0.9175



원본과 Hyperbolic 결과 비교 플롯
정확도 0.83



원본과 Gaussian RBF 결과 비교 플롯
정확도 0.9975

RBF 가 가장 정확도가 높다.