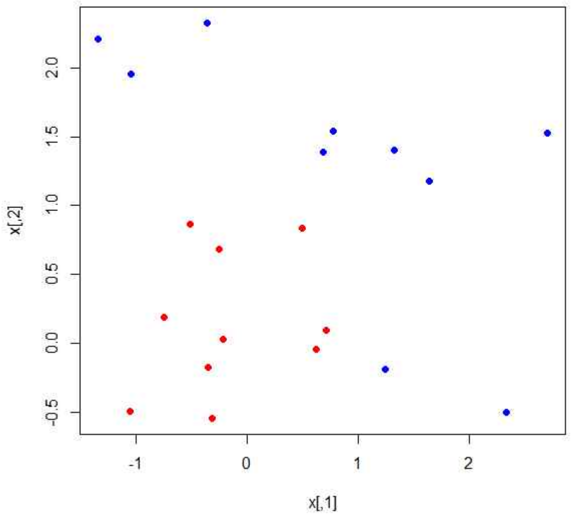
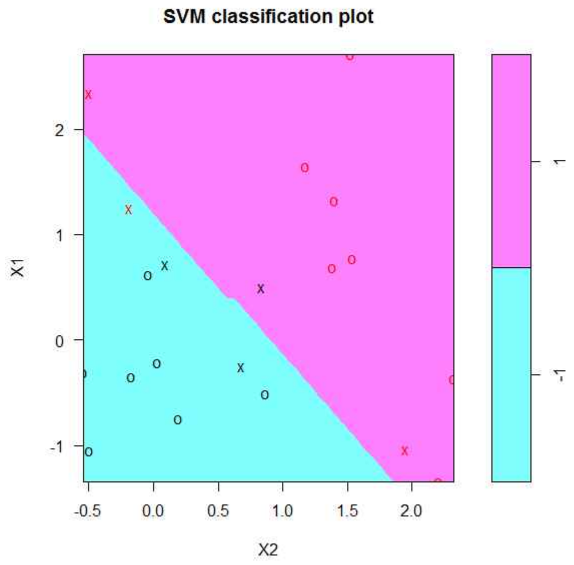
데이터 사이언스 과제5

< 1. Support Vector Machines (SVM) >

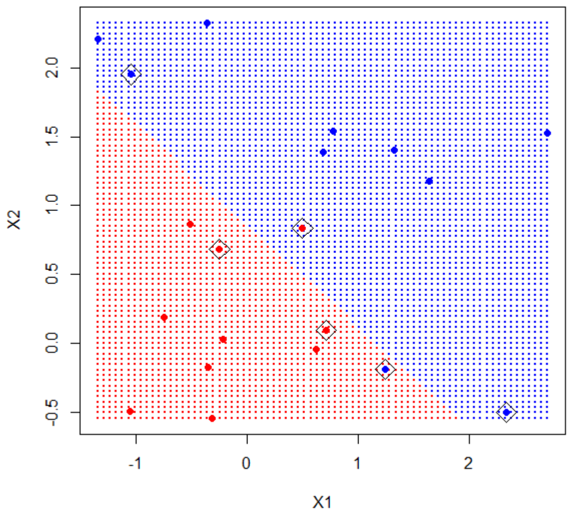
```
> #1) Support Vector Machines Algorithm
>
> #2) Support Vector Machines in R
> set.seed(10111)
> x = matrix(rnorm(40), 20, 2)
> y = rep(c(-1, 1), c(10, 10))
> x[y == 1,] = x[y == 1,] + 1
> plot(x, col = y + 3, pch = 19)
```



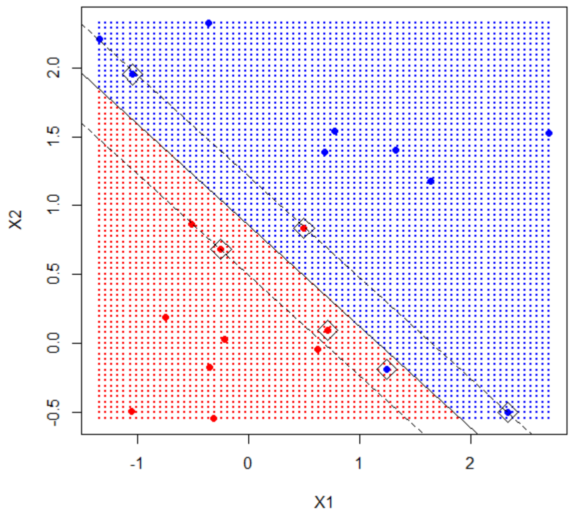
```
> #install.packages("e1071")
> library(e1071)
경고메시지(들):
패키지 'e1071'는 R 버전 3.5.1에서 작성되었습니다
>
> dat = data.frame(x, y = as.factor(y))
> svmfit = svm(y ~ ., data = dat, kernel = "linear", cost = 10, scale =
FALSE)
> print(svmfit)

Call:
svm(formula = y ~ ., data = dat, kernel = "linear", cost = 10,
    scale = FALSE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  10
      gamma:  0.5

Number of Support Vectors:  6

>
> plot(svmfit, dat)
```



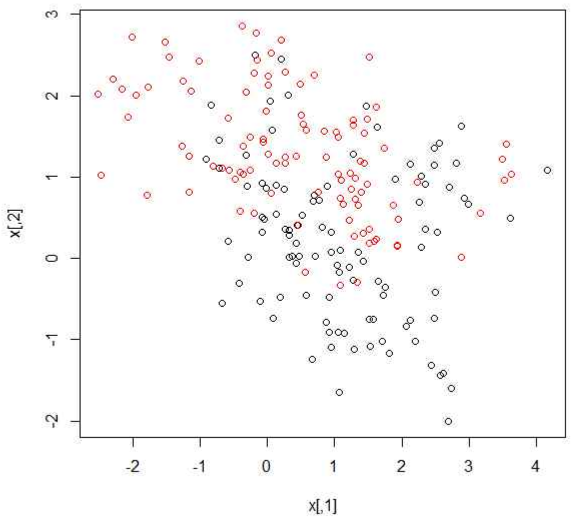
```
> make.grid = function(x, n = 75) {
+   grange = apply(x, 2, range)
+   x1 = seq(from = grange[1,1], to = grange[2,1], length = n)
+   x2 = seq(from = grange[1,2], to = grange[2,2], length = n)
+   expand.grid(X1 = x1, X2 = x2)
+ }
>
> xgrid = make.grid(x)
> xgrid[1:10,]
          X1         X2
1  -1.3406379 -0.5400074
2  -1.2859572 -0.5400074
3  -1.2312766 -0.5400074
4  -1.1765959 -0.5400074
5  -1.1219153 -0.5400074
6  -1.0672346 -0.5400074
7  -1.0125540 -0.5400074
8  -0.9578733 -0.5400074
9  -0.9031927 -0.5400074
10 -0.8485120 -0.5400074
>
> ygrid = predict(svmfit, xgrid)
> plot(xgrid, col = c("red","blue")[as.numeric(ygrid)], pch = 20, cex = .2)
> points(x, col = y + 3, pch = 19)
> points(x[svmfit$index,], pch = 5, cex = 2)
```



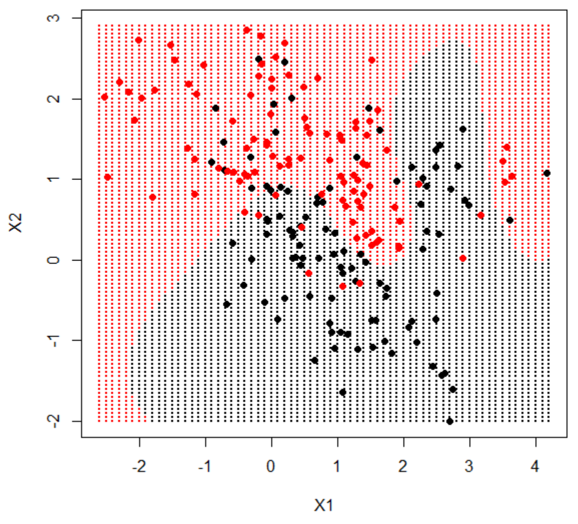
```
> beta = drop(t(svmfit$coefs)%*%x[svmfit$index,])
> beta0 = svmfit$rho
>
> plot(xgrid, col = c("red", "blue")[as.numeric(ygrid)], pch = 20, cex = .2)
> points(x, col = y + 3, pch = 19)
> points(x[svmfit$index,], pch = 5, cex = 2)
> abline(beta0 / beta[2], -beta[1] / beta[2])
> abline((beta0 - 1) / beta[2], -beta[1] / beta[2], lty = 2)
> abline((beta0 + 1) / beta[2], -beta[1] / beta[2], lty = 2)
```



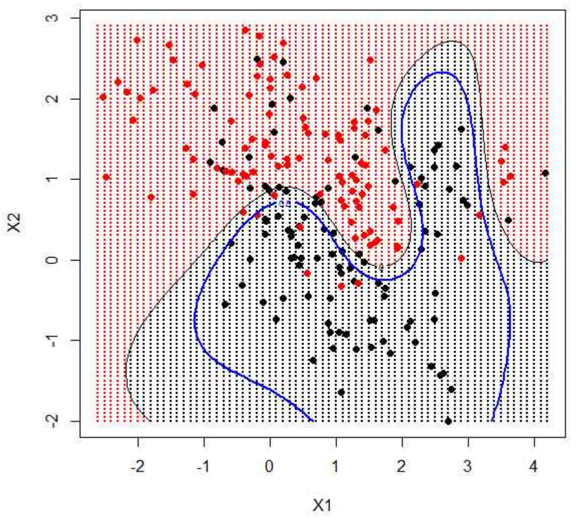
```
> load(file = "ESL.mixture.rda")
> names(ESL.mixture)
[1] "x"          "y"          "xnew"       "prob"       "marginal" "px1"
[7] "px2"        "means"
>
> rm(x, y)
> attach(ESL.mixture)
>
> plot(x, col = y + 1)
```



```
> dat = data.frame(y = factor(y), x)
> fit = svm(factor(y) ~ ., data = dat, scale = FALSE, kernel = "radial", cost
= 5)
>
> xgrid = expand.grid(X1 = px1, X2 = px2)
> ygrid = predict(fit, xgrid)
>
> plot(xgrid, col = as.numeric(ygrid), pch = 20, cex = .2)
> points(x, col = y + 1, pch = 19)
```



```
> func = predict(fit, xgrid, decision.values = TRUE)
> func = attributes(func)$decision
> xgrid = expand.grid(X1 = px1, X2 = px2)
> ygrid = predict(fit, xgrid)
> plot(xgrid, col = as.numeric(ygrid), pch = 20, cex = .2)
> points(x, col = y + 1, pch = 19)
> contour(px1, px2, matrix(func, 69, 99), level = 0, add = TRUE)
> contour(px1, px2, matrix(func,69,99), level = 0.5, add = TRUE, col =
"blue", lwd = 2)
```



X2
X1
-2
-1
0
1
2
3
4