Natan Alper 4/28/2020

Business Intelligence & Consumer Insights- Professor Kovtun

HW #7

**1**

> # 1

> ci <- data.frame(cps\_income)

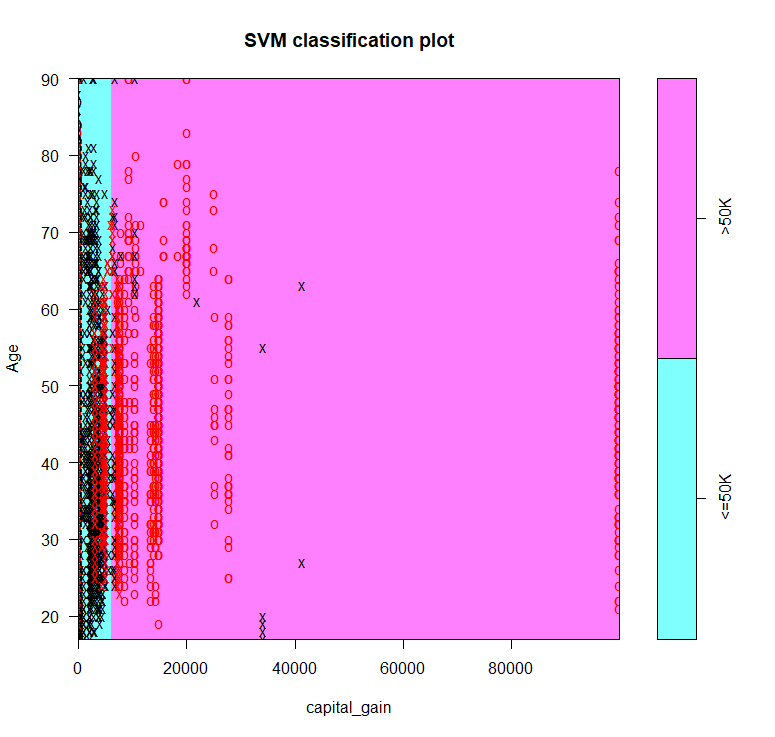
> # Select only quant columns (last col is INCOME)

> ciQuant <- ci[, c(1,3,5,11,12,13,15)]

> library(e1071)

> model=svm(Income~.,data=ciQuant,kernel="linear",type="C-classification",cost=1)

> plot(model,data=ciQuant,formula=Age~capital\_gain)



**2**

> # 2

> # Run randsamp only once

> randSamp <- sample((1:nrow(ci)), size = 30000)

> training <- ciQuant[randSamp, ]

> testing <- ciQuant[-randSamp, ]

> model=svm(Income~.,data=training,kernel="linear",type="C-classification",cost=1)

> predictions= predict(model, newdata = testing)

> misclass=sum(testing$Income!=predictions)/length(predictions)

> misclass

[1] 0.1996094

**3**

> # 3

> model1=svm(Income~.,data=training,kernel="radial",type="C-classification",cost=1)

> predictions1= predict(model1, newdata = testing)

> misclass1=sum(testing$Income!=predictions1)/length(predictions1)

> misclass1

[1] 0.1777344

> model2=svm(Income~.,data=training,kernel="linear",type="C-classification",cost=10)

> predictions2= predict(model2, newdata = testing)

> misclass2=sum(testing$Income!=predictions2)/length(predictions2)

> misclass2

[1] 0.1996094

### Which model did better?

## HW6 Q1e misclass = 0.203125

## HW6 Q2 misclass = 0.1824219

## HW7 Q2 misclass = 0.1996094

## HW7 Q3 model1 misclass = 0.1777344 - new best

## HW7 Q3 model2 misclass = 0.1996094

**4**

> # 4

> # Only run samp code once

> samp=sample((1:32560), 32560, replace = FALSE)

> # We create 5 folds

> g1=samp[1:6512]

> g2=samp[6513:13024]

> g3=samp[13025:19536]

> g4=samp[19537:26048]

> g5=samp[26051:32560]

> g5=samp[26049:32560]

> Groups <- data.frame(g1,g2,g3,g4,g5)

> preds <- c()

> for(i in 1:5){

+ model1 <- svm(Income~.,data=ciQuant[-Groups[,i],],kernel="radial",type="C-classification",cost=1)

+ predictions\_per\_fold <- predict(model1,type="class",newdata=ciQuant[Groups[,i],])

+ preds <- c(preds,as.character(predictions\_per\_fold))

+ }

> misclass\_cross <- sum(ciQuant$Income!=preds)/length(preds)

> misclass\_cross

[1] 0.2982801

**5**

> # 5a

> FPR=c()

> FNR=c()

> model=svm(Income~.,data=training,kernel="linear",type="C-classification",cost=1)

> predictions= predict(model, newdata = testing)

> # If we want to compute the FPR for "<=50K" class then we need to see the percentage of ">50K" individuals that are predicted to be "<=50K":

> over50obs=(1:2560)[testing$Income==" >50K"]

> FPR=sum(testing$Income[over50obs]!=predictions[over50obs])/length(predictions[over50obs])

> # If we want to compute the FNR for "<=50K" class then we need to see the percentage of "<=50K" individuals that are predicted to be ">50K":

> under50obs=(1:2560)[testing$Income==" <=50K"]

> FNR=sum(testing$Income[under50obs]!=predictions[under50obs])/length(predictions[under50obs])

> FPR

[1] 0.686901

> FNR

[1] 0.04188211

> # 5b

> # If we want to compute the FPR for ">50K" class then we need to see the percentage of "<=50K" individuals that are predicted to be ">50K":

> under50obs=(1:2560)[testing$Income==" <=50K"]

> FPR=sum(testing$Income[under50obs]!=predictions[under50obs])/length(predictions[under50obs])

> # If we want to compute the FNR for ">50K" class then we need to see the percentage of ">50K" individuals that are predicted to be "<=50K":

> over50obs=(1:2560)[testing$Income==" >50K"]

> FNR=sum(testing$Income[over50obs]!=predictions[over50obs])/length(predictions[over50obs])

> FPR

[1] 0.04188211

> FNR

[1] 0.686901