breastCancer2.r

setwd("C:/Users/Tony/Dropbox/Rowan/DM2/Lecture1/WisconsinBreastCancer")  
f=read.csv(file="wdbc\_data\_abclean.csv",header=F,stringsAsFactors = TRUE)  
  
dim(f)

## [1] 569 32

f[1,]

## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11  
## 1 842302 1 17.99 10.38 122.8 1001 0.1184 0.2776 0.3001 0.1471 0.2419  
## V12 V13 V14 V15 V16 V17 V18 V19 V20  
## 1 0.07871 1.095 0.9053 8.589 153.4 0.006399 0.04904 0.05373 0.01587  
## V21 V22 V23 V24 V25 V26 V27 V28 V29 V30  
## 1 0.03003 0.006193 25.38 17.33 184.6 2019 0.1622 0.6656 0.7119 0.2654  
## V31 V32  
## 1 0.4601 0.1189

## We don't need the first column, it's just an id  
ff <-subset(f,select=-V1)  
dim(ff)

## [1] 569 31

library(plyr)  
count(ff,c("V2"))

## V2 freq  
## 1 0 357  
## 2 1 212

## 357 benign, 212 malignant  
  
  
set.seed(2)  
train=sample(1:nrow(ff),nrow(ff)\*(8/10))  
test=-train  
  
trainingData=ff[train,]  
testingData=ff[test,]  
dim(trainingData)

## [1] 455 31

dim(testingData)

## [1] 114 31

library("neuralnet")  
  
formula = V2~V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24+V25+V26+V27+V28+V29+V30+V31+V32  
  
nnet<-neuralnet(formula,trainingData, hidden=32, threshold=0.1)  
results<-compute(nnet,testingData[,2:31])  
testingData$result<- sapply(results$net.result, function(b) {  
 if (b<=.5){  
 return(0)  
 }else{  
 return(1)  
 }})  
count(testingData,c("V2","result"))

## V2 result freq  
## 1 0 0 76  
## 2 0 1 2  
## 3 1 0 3  
## 4 1 1 33

err = 5/(5+76+33)  
err

## [1] 0.04385964912

library(e1071)  
  
## We make V2 a factor becuase if it's a factor the SVM call  
## solves as classification. Otherwise it attempts a regression  
## in other words it will return a 0 or 1 in this case rather than a range between 0 and 1  
formula = as.factor(V2)~V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17+V18+V19+V20+V21+V22+V23+V24+V25+V26+V27+V28+V29+V30+V31+V32  
  
svm\_model <- svm(formula, data=trainingData)  
summary(svm\_model)

##   
## Call:  
## svm(formula = formula, data = trainingData)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.03333333333   
##   
## Number of Support Vectors: 104  
##   
## ( 52 52 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## 0 1

testingData$result <- predict(svm\_model,testingData)  
count(testingData,c("V2","result"))

## V2 result freq  
## 1 0 0 78  
## 2 1 0 2  
## 3 1 1 34

err = 2/(2+78+34)  
err

## [1] 0.01754385965

trainingData[1,]

## V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12  
## 106 1 13.11 15.56 87.21 530.2 0.1398 0.1765 0.2071 0.09601 0.1925 0.07692  
## V13 V14 V15 V16 V17 V18 V19 V20 V21  
## 106 0.3908 0.9238 2.41 34.66 0.007162 0.02912 0.05473 0.01388 0.01547  
## V22 V23 V24 V25 V26 V27 V28 V29 V30 V31  
## 106 0.007098 16.31 22.4 106.4 827.2 0.1862 0.4099 0.6376 0.1986 0.3147  
## V32  
## 106 0.1405

x <- subset(trainingData, select=-V2)  
y <- as.factor(trainingData$V2)  
tail(y)

## [1] 0 1 0 1 0 1  
## Levels: 0 1

svm\_model1 <- svm(x,y)  
summary(svm\_model1)

##   
## Call:  
## svm.default(x = x, y = y)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.03333333333   
##   
## Number of Support Vectors: 104  
##   
## ( 52 52 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## 0 1

xTest <-subset(testingData, select=c(-V2,-result))  
yTest <- testingData$V2  
pred <- predict(svm\_model1,xTest)  
  
table(pred,yTest)

## yTest  
## pred 0 1  
## 0 78 2  
## 1 0 34

err = 2/(2+78+34)  
err

## [1] 0.01754385965

svm\_tune <- tune(svm, train.x=x, train.y=y,   
 kernel="radial", ranges=list(cost=10^(-1:2), gamma=c(.5,1,2)))  
print(svm\_tune)

##   
## Parameter tuning of 'svm':  
##   
## - sampling method: 10-fold cross validation   
##   
## - best parameters:  
## cost gamma  
## 10 0.5  
##   
## - best performance: 0.1692270531

svm\_model\_after\_tune <- svm(as.factor(V2) ~ ., trainingData, kernel="radial", cost=10^(-1:1), gamma=0.01)  
summary(svm\_model\_after\_tune)

##   
## Call:  
## svm(formula = as.factor(V2) ~ ., data = trainingData, kernel = "radial",   
## cost = 10^(-1:1), gamma = 0.01)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 0.1 1 10   
## gamma: 0.01   
##   
## Number of Support Vectors: 210  
##   
## ( 104 106 )  
##   
##   
## Number of Classes: 2   
##   
## Levels:   
## 0 1

pred <- predict(svm\_model\_after\_tune,xTest)  
table(pred,yTest)

## yTest  
## pred 0 1  
## 0 78 4  
## 1 0 32

## Tuning made it slightly worse; still better than Neural Net