breastCancerROC.r

library(plyr)  
  
  
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

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  
  
  
## We saw a few weeks ago we only need V26 and V30  
formula = V2~V26+V30  
newF <- ff[,c(1,25,29)]  
newF[1:3,]

## V2 V26 V30  
## 1 1 2019 0.2654  
## 2 1 1956 0.1860  
## 3 1 1709 0.2430

maxs <- apply(newF, 2, max)  
mins <- apply(newF, 2, min)  
scaledF <- as.data.frame(scale(newF, center = mins, scale = maxs - mins))  
  
  
set.seed(2)  
train=sample(1:nrow(scaledF),nrow(scaledF)\*(8/10))  
test=-train  
  
trainingData=scaledF[train,]  
testingData=scaledF[test,]  
  
  
library(e1071)  
  
formula = V2~V26+V30  
  
svm\_model <- svm(formula, data=trainingData)  
summary(svm\_model)

##   
## Call:  
## svm(formula = formula, data = trainingData)  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.5   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 149

testingData$result <- predict(svm\_model,testingData)  
  
trainingData[1,]

## V2 V26 V30  
## 106 1 0.1577861 0.6824742

#install.packages("pROC")  
library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

roc\_SVM <- plot(roc(testingData$V2, testingData$result), print.auc = TRUE, col = "blue", main="SVM-blue, NN-Green")  
  
  
  
library("neuralnet")  
  
nnet<-neuralnet(formula,trainingData, hidden=6, threshold=0.1)  
results<-compute(nnet,testingData[,2:3])  
  
roc\_NN <- plot(roc(testingData$V2, results$net.result), print.auc = TRUE,   
 col = "green", print.auc.y = .4, add = TRUE)

## Warning in roc.default(testingData$V2, results$net.result): Deprecated use  
## a matrix as predictor. Unexpected results may be produced, please pass a  
## numeric vector.

## We might as well add a random predictor  
count(trainingData$V2)

## x freq  
## 1 0 279  
## 2 1 176

279+176

## [1] 455

rr <- function(){  
 if (runif(1) > 279/455)  
 return(1)  
 else  
 return(0)  
}  
  
rr()

## [1] 1

testingData[1,]

## V2 V26 V30 result  
## 2 1 0.4352143138 0.6391752577 0.9616907555

testingData$randRes <- 0  
for (i in 1:dim(testingData)[2])  
 testingData[i,5] <- rr()  
  
roc\_rand <- plot(roc(testingData$V2, testingData$randRes), print.auc = TRUE,   
 col = "red", print.auc.y = .8, add = TRUE)

