Assignment 3
Vikram Singh Chandel
Roll No.: **173040020**

Reading and partioning data :

```
rm(list=ls())
set.seed(1)
setwd('Dropbox/MTech/Hydroinformatics/Assignment 3/')
X=t(read.csv('cancerInputs.csv', header = FALSE))
Y=t(read.csv('cancerTargets.csv', header = FALSE))
#install.packages("tree")
require("tree")
#dev.new()
Xtrain=X[1:549, ]
Xtest=X[550:dim(X)[1], ]
# 1 is Benign
Ytrain=factor(Y[1:549,1])
Ytest=factor(Y[550:dim(Y)[1],1])

dfr = data.frame(Xtrain,"Benign"=Ytrain)
dftest = data.frame(Xtest,"Benign"=Ytest)
```

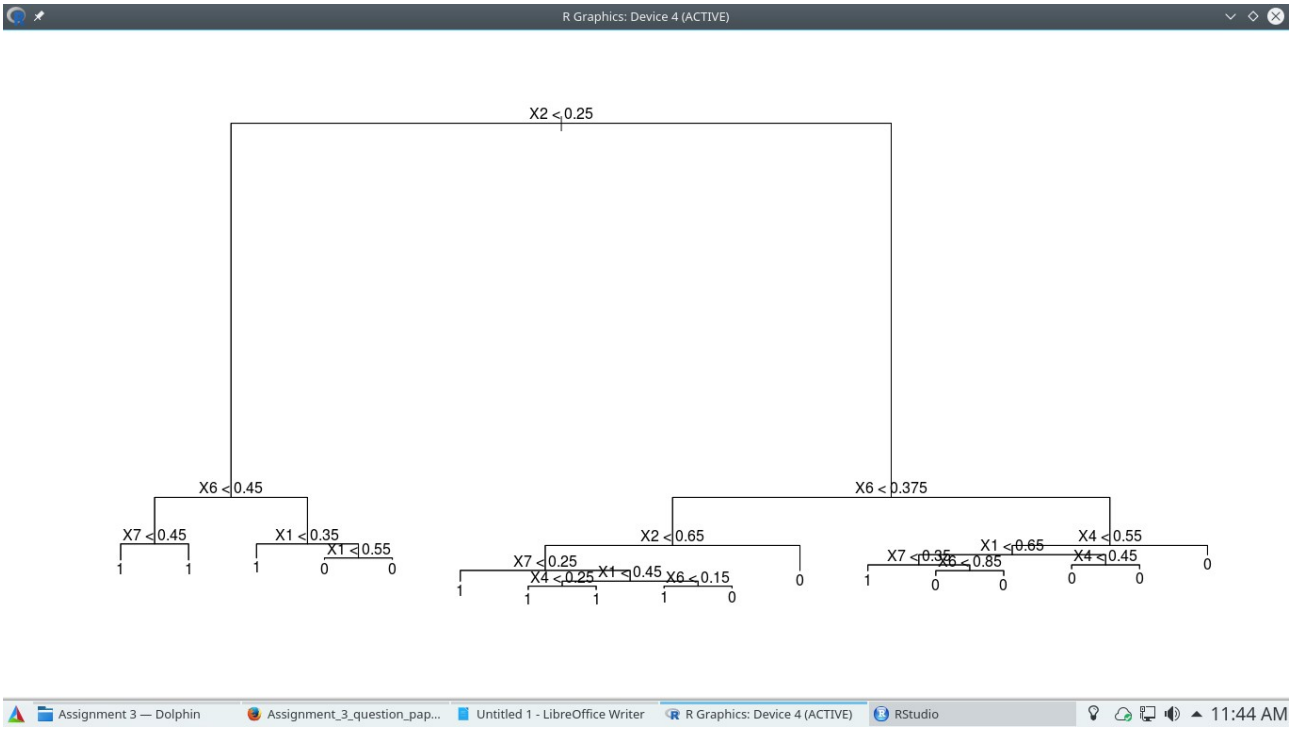
A. CART\Bagging

CART

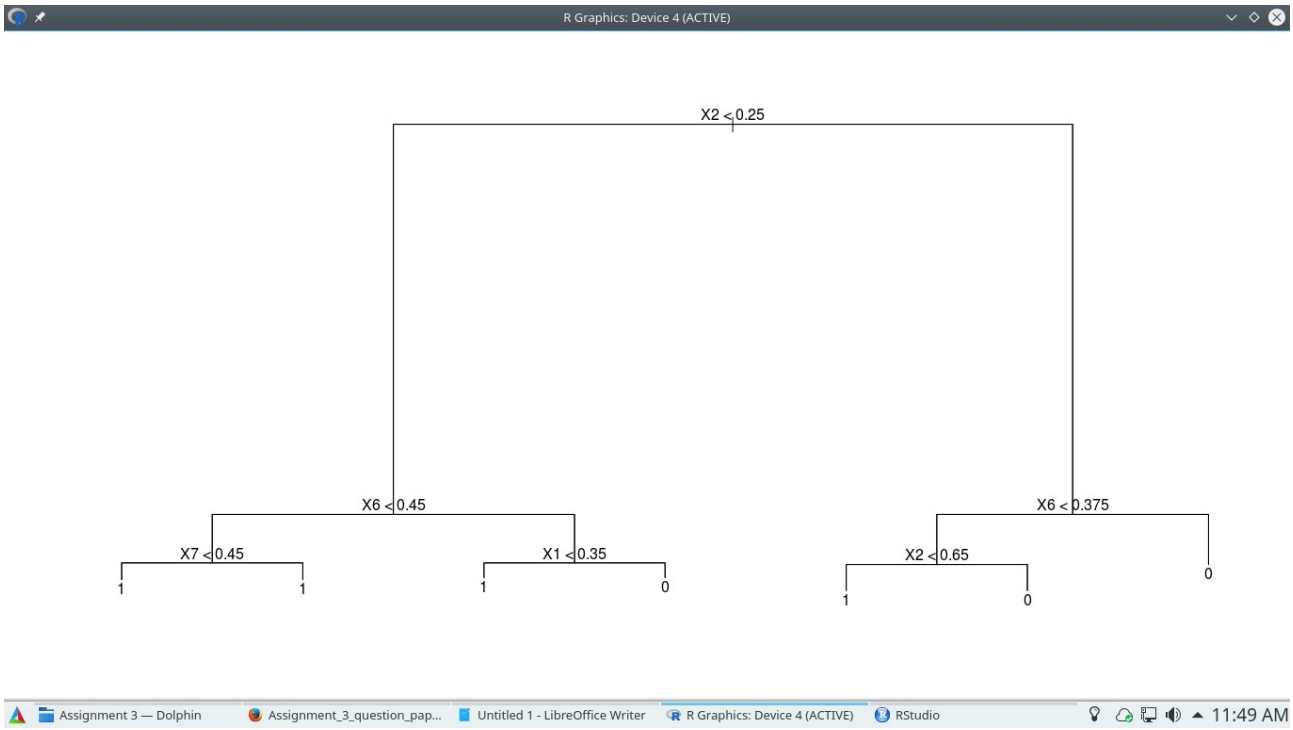
```
TR=tree(Benign ~ . ,data=dfr,mindev=0.001)

plot(TR)
text(TR,pretty=0)

#cross validation with 10 sample points giving k=55
cree=cv.tree(TR,FUN=prune.tree,K=55)

#pruning
pruned=prune.tree(TR,best=7)
# 7 number of terminal nodes was selected from output of cross validation

#plotting pruned tree
plot(pruned)
text(pruned,pretty=0)

#prediction on test data
tree.pred=predict(pruned,dftest,type=c("class"))
#confusion matrix
Tab=table(tree.pred,Ytest)
```

Misclassification Rate is 4 % on test dataset

Full Depth Tree



Pruned Tree



Bagging:
As bagging is a special case of random forest, 'randomForest' package was used.

```
#Bagging
#install.packages('randomForest')
require(randomForest)
BAG=randomForest(Benign ~ .,data=dfr , mtry=9,ntree=1000)
bagpred=predict(BAG,dftest)
Tab=table(bagpred,Ytest)
```

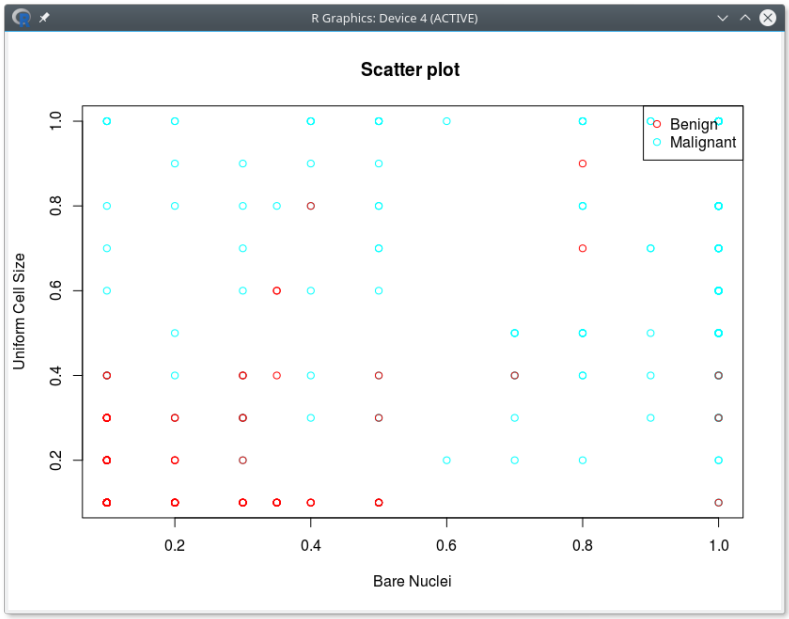
Misclassification Rate is 2% on test dataset

B) SVC/SVM

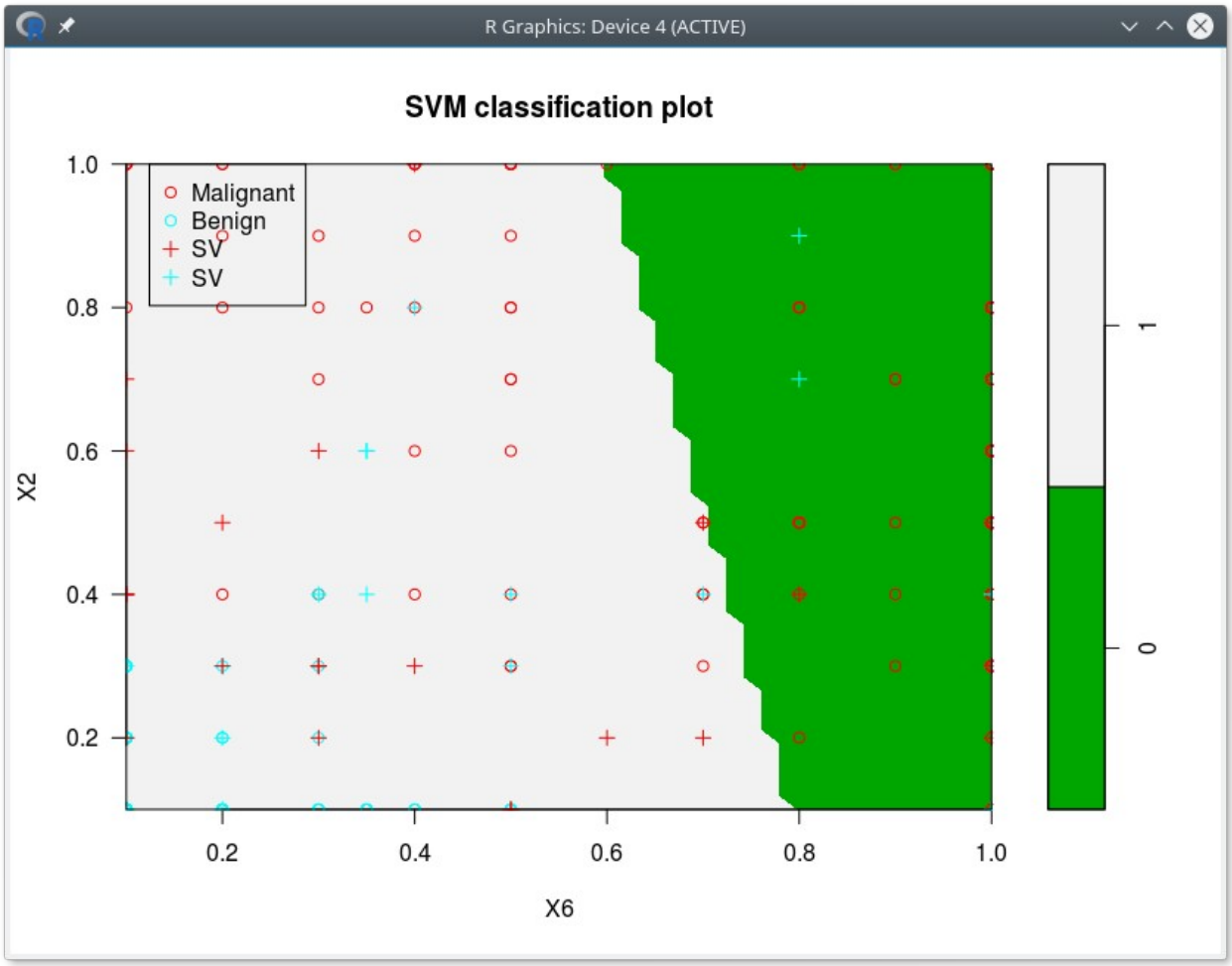
SVC

```
#install.packages('e1071')
require(e1071)

svmfi=svm(Benign~.,data=dfr, kernel="linear", cost=1000,scale=FALSE)

#tuning cost
tun=tune(svm,Benign~.,data=dfr,kernel="linear",ranges=list(cost=c(0.001, 0.01, 0.1, 1,5,10,100)))

#cost =5  gave best performance
svmfi=svm(Benign~.,data=dfr, kernel="linear", cost=5,scale=FALSE)
svmpred=predict(svmfi,dftest)
Tab=table(svmpred,Ytest)

#plotting
colo=rainbow(2);
a=which(dfr$Benign==0)
plot(dfr$X6[a],dfr$X2[a],type="p",col=colo[2],xlab ="Bare Nuclei" , ylab ="Uniform Cell Size" ,
    main = "Scatter plot ")
a=which(dfr$Benign==1)
points(dfr$X6[a],dfr$X2[a],type="p",col=colo[1])
colo=rainbow(2);
legend('topright',c('Benign','Malignant'),pch=c(1,1),col=c(colo[1],colo[2]))
plot(svmfi,dfr,X2~X6,slice=list(X1=1),svSymbol =3, dataSymbol = 1, symbolPalette =colo,
    color.palette = terrain.colors)

legend('topleft',c('Malignant','Benign','SV','SV'),pch=c(1,1,3,3),col=colo)
```

Misclassification Rate is 2 % on test dataset



Scatter plot Uniform Cell thickness vs Bare Nuclei

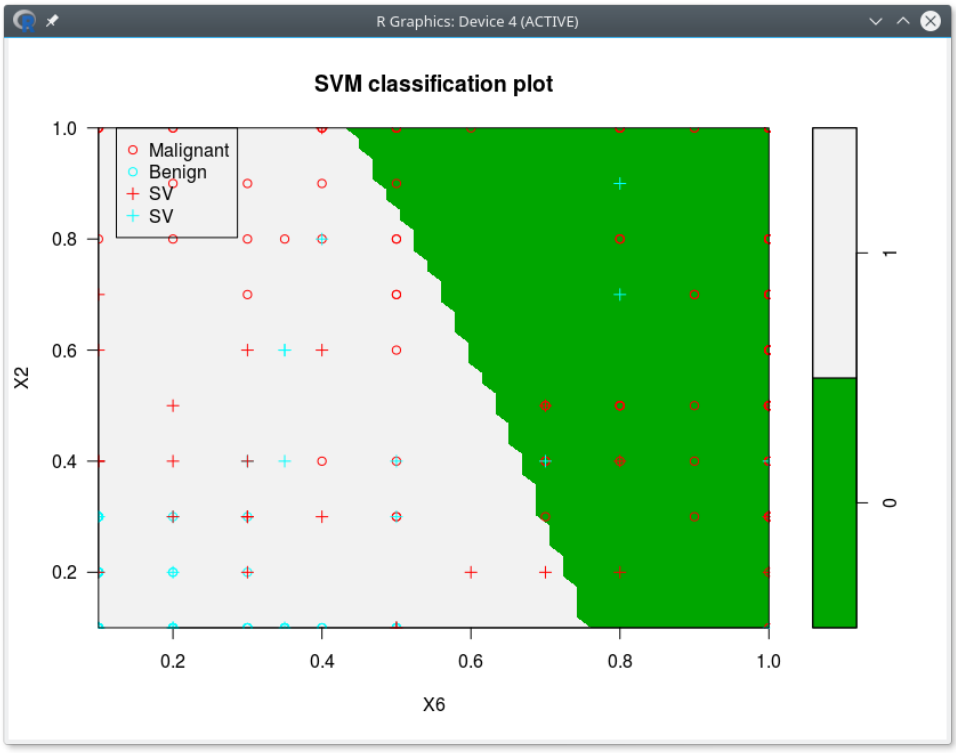


SVC classification plot
(Plot for Clump thickness =1)

Support Vector Machine

```
#SVM
svmfi=svm(Benign~.,data=dfr, kernel="radial", cost=1000,scale=FALSE)
tun=tune(svm,Benign~.,data=dfr,kernel="radial",ranges=list(cost=c(0.001, 0.01, 0.1, 1,5,10,100)))
svmfi=svm(Benign~.,data=dfr, kernel="radial", cost=5,scale=FALSE)
svmpred=predict(svmfi,dftest)
Tab=table(svmpred,Ytest)

plot(svmfi,dfr,X2~X6,slice=list(X1=1),svSymbol =3, dataSymbol = 1, symbolPalette =colo,
    color.palette = terrain.colors)
legend('topleft',c('Malignant','Benign','SV','SV'),pch=c(1,1,3,3),col=colo)
```

Misclassification Rate is 2% on test dataset



SVM Classification plot
(Plot for Clump thickness =1)

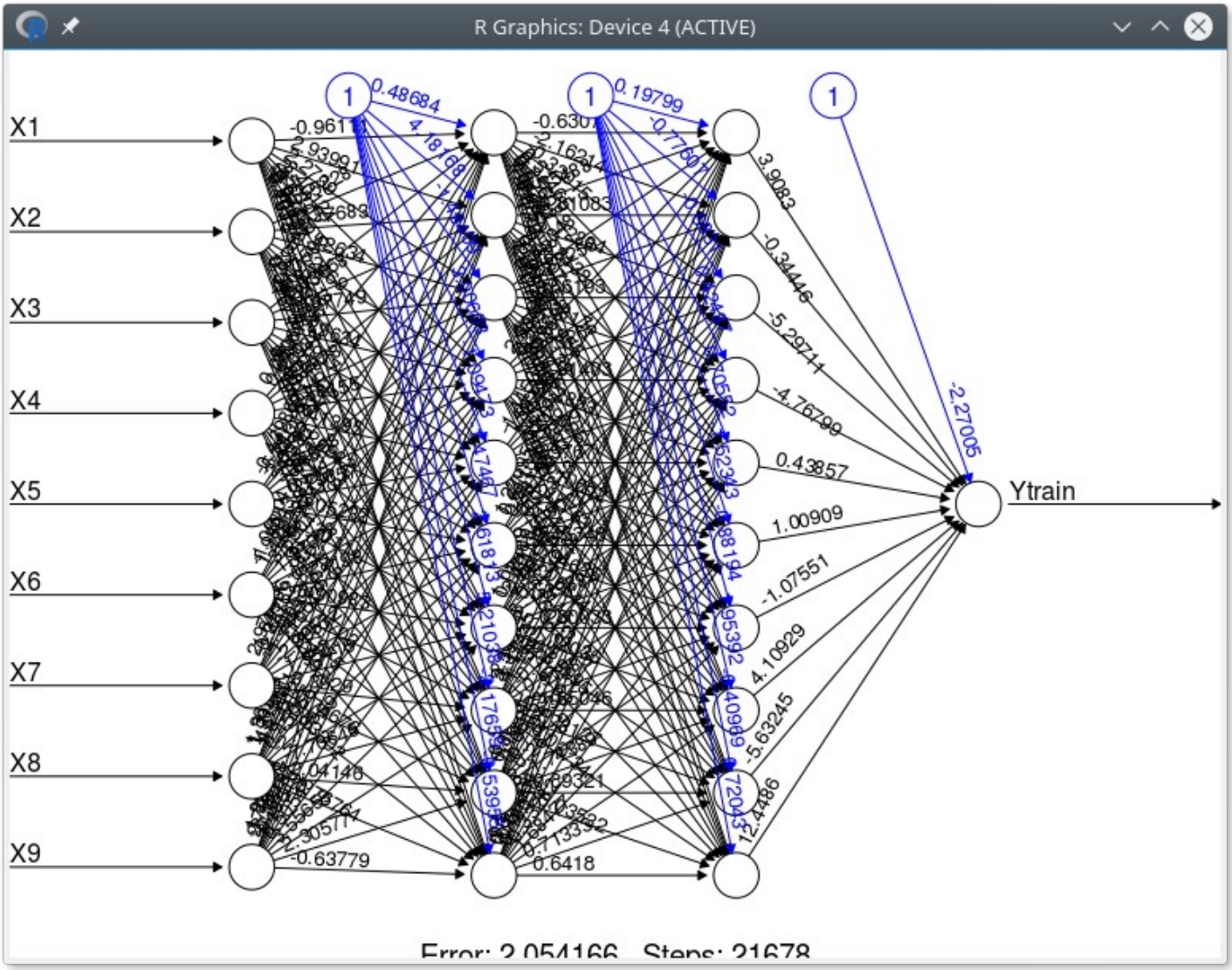
The performance of Support Vector Classifier is same as Support Vector Machine because decision boundary in SVM is nearly linear hence predicts similar to SVC.

ANN
Training Algorithm: Gradient Decent with learning parameter 0.01
Transfer Function : Sigmoid

```
#install.packages('neuralnet')
require(neuralnet)
Ytrain=Y[1:549,1]
nndel=neuralnet(Ytrain ~ X1 + X2 +X3 + X4 +X5 + X6 +X7 + X8 +X9 ,data=dfr,
                hidden=c(10,10),act.fct = 'logistic',  algorithm='backprop', learningrate =0.01,
linear.output = FALSE)

plot(nndel)
prednn=compute(nndel,Xtest)
names(prednn)

predict=as.factor(ifelse(prednn$net.result>=0.5,1,0))
table(predict,Ytest)
mcr=3/1.5

intpre=prednn$net.result
Ytest=Y[550:dim(Y)[1],1]
MSE=sum(((intpre)-(Ytest))^2)/length(Ytest)
```

Misclassification Rate is 2% on test dataset
Mean Squared Error = 0.01713

	Misclassification Rate
Bagged Tree	2 %
SVM	2 %
ANN	2 %
Performance of Bagged Tree, SVM and ANN are same on the test dataset.	


R Graphics: Device 4 (ACTIVE)
X1
X2
X3
X4
X5
X6
X7
X8
X9
1
0.48684
0.19799
-0.63779
0.6418
3.9083
-0.34446
-5.29711
-4.76799
0.43857
1.00909
-1.07551
4.10929
-5.63245
12.4486
-2.27005
Ytrain
Error: 2.054166 Steps: 21678