Project2

Rex Cory

3/10/2021

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(plyr)

## Warning: package 'plyr' was built under R version 4.0.3

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.0.3

# library(class)##Keep  
library(e1071)

## Warning: package 'e1071' was built under R version 4.0.3

# library(FNN) ##Keep  
# library(gmodels) ##Keep  
# library(psych)##May Need  
library(klaR)

## Warning: package 'klaR' was built under R version 4.0.4

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 4.0.3

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(nnet)  
library(MASS)  
library(rpart)  
library(mlbench)

## Warning: package 'mlbench' was built under R version 4.0.4

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.0.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

## The following object is masked from 'package:dplyr':  
##   
## combine

library(party)

## Warning: package 'party' was built under R version 4.0.4

## Loading required package: grid

## Loading required package: mvtnorm

## Warning: package 'mvtnorm' was built under R version 4.0.3

## Loading required package: modeltools

## Warning: package 'modeltools' was built under R version 4.0.3

## Loading required package: stats4

##   
## Attaching package: 'modeltools'

## The following object is masked from 'package:plyr':  
##   
## empty

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.0.4

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.0.3

library(ipred)

## Warning: package 'ipred' was built under R version 4.0.4

library(ROCR)

## Warning: package 'ROCR' was built under R version 4.0.4

data(BreastCancer)

# remove missing values  
BreastCancer <- na.omit(BreastCancer)   
# remove the unique identifier  
BreastCancer$Id <- NULL  
# partition the data set for 60% training and 40% evaluation   
set.seed(1234)   
smp\_size <- floor(0.6 \* nrow(BreastCancer))  
train\_ind <- sample(seq\_len(nrow(BreastCancer)), size = smp\_size)  
train.df <- BreastCancer[train\_ind, ]  
eval.df <- BreastCancer[-train\_ind, ]

mysvm <- svm(Class ~ ., train.df)  
mysvm.pred <- predict(mysvm, eval.df)

mynb <- NaiveBayes(Class ~ ., train.df)  
mynb.pred <- predict(mynb,eval.df)

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 1

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 4

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 8

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 11

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 13

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 18

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 20

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 24

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 25

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 26

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 27

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 28

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 32

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 37

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 43

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 44

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 61

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 63

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 64

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 68

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 71

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 73

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 84

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 85

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 87

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 90

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 92

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 93

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 95

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 96

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 101

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 102

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 104

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 106

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 107

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 108

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 110

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 118

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 124

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 131

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 132

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 135

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 141

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 143

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 144

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 153

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 155

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 162

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 163

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 164

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 165

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 173

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 174

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 181

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 182

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 183

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 190

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 191

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 196

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 202

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 216

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 218

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 226

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 238

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 239

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 250

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 258

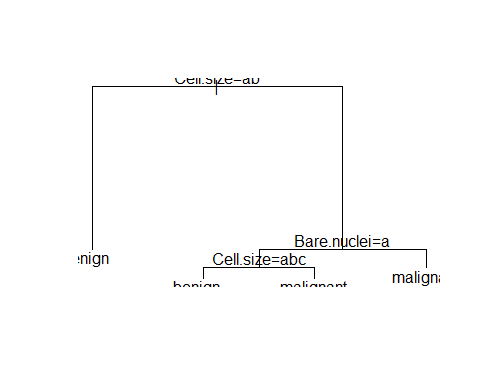
## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 274

mynnet <- nnet(Class ~ ., train.df, size=1)

## # weights: 83  
## initial value 280.286547   
## iter 10 value 16.065766  
## iter 20 value 0.233822  
## iter 30 value 0.008314  
## iter 40 value 0.001200  
## iter 50 value 0.000242  
## final value 0.000085   
## converged

mynnet.pred <- predict(mynnet,eval.df,type="class")

mytree <- rpart(Class ~ ., train.df)  
plot(mytree); text(mytree) # in "iris\_tree.ps"



summary(mytree)

## Call:  
## rpart(formula = Class ~ ., data = train.df)  
## n= 409   
##   
## CP nsplit rel error xerror xstd  
## 1 0.8040541 0 1.0000000 1.0000000 0.06566406  
## 2 0.0472973 1 0.1959459 0.2229730 0.03721581  
## 3 0.0100000 3 0.1013514 0.1554054 0.03147996  
##   
## Variable importance  
## Cell.size Cell.shape Epith.c.size Bare.nuclei Bl.cromatin   
## 21 18 16 15 15   
## Normal.nucleoli Marg.adhesion   
## 14 1   
##   
## Node number 1: 409 observations, complexity param=0.8040541  
## predicted class=benign expected loss=0.3618582 P(node) =1  
## class counts: 261 148  
## probabilities: 0.638 0.362   
## left son=2 (242 obs) right son=3 (167 obs)  
## Primary splits:  
## Cell.size splits as LLRRRRRRRR, improve=137.9948, (0 missing)  
## Cell.shape splits as LLLRRRRRRR, improve=131.6266, (0 missing)  
## Epith.c.size splits as LLRRRRRRRR, improve=122.0599, (0 missing)  
## Bl.cromatin splits as LLLRRRRRRR, improve=119.9556, (0 missing)  
## Bare.nuclei splits as LLRRRRRRRR, improve=119.7315, (0 missing)  
## Surrogate splits:  
## Cell.shape splits as LLRRRRRRRR, agree=0.932, adj=0.832, (0 split)  
## Epith.c.size splits as LLRRRRRRRR, agree=0.910, adj=0.778, (0 split)  
## Bl.cromatin splits as LLLRRRRRRR, agree=0.875, adj=0.695, (0 split)  
## Bare.nuclei splits as LRRRRRRRRR, agree=0.866, adj=0.671, (0 split)  
## Normal.nucleoli splits as LLRRRRRRRR, agree=0.866, adj=0.671, (0 split)  
##   
## Node number 2: 242 observations  
## predicted class=benign expected loss=0.02066116 P(node) =0.591687  
## class counts: 237 5  
## probabilities: 0.979 0.021   
##   
## Node number 3: 167 observations, complexity param=0.0472973  
## predicted class=malignant expected loss=0.1437126 P(node) =0.408313  
## class counts: 24 143  
## probabilities: 0.144 0.856   
## left son=6 (24 obs) right son=7 (143 obs)  
## Primary splits:  
## Bare.nuclei splits as LRRRRRRRRR, improve=12.984660, (0 missing)  
## Cell.size splits as LLLRRRRRRR, improve=10.358510, (0 missing)  
## Cell.shape splits as LLLRRRRRRR, improve= 9.792951, (0 missing)  
## Marg.adhesion splits as LLRRRRRRRR, improve= 8.414435, (0 missing)  
## Bl.cromatin splits as LLRRRRRRRR, improve= 8.197823, (0 missing)  
## Surrogate splits:  
## Cell.shape splits as LRRRRRRRRR, agree=0.886, adj=0.208, (0 split)  
## Bl.cromatin splits as LRRRRRRRRR, agree=0.886, adj=0.208, (0 split)  
##   
## Node number 6: 24 observations, complexity param=0.0472973  
## predicted class=benign expected loss=0.375 P(node) =0.05867971  
## class counts: 15 9  
## probabilities: 0.625 0.375   
## left son=12 (14 obs) right son=13 (10 obs)  
## Primary splits:  
## Cell.size splits as LLLRRRRRRR, improve=9.450000, (0 missing)  
## Epith.c.size splits as LLLRRRRRRR, improve=7.977273, (0 missing)  
## Marg.adhesion splits as LLLLRRRRRR, improve=7.720588, (0 missing)  
## Cell.shape splits as LLLRRRRRRR, improve=6.750000, (0 missing)  
## Bl.cromatin splits as LLLL--RR-R, improve=6.000000, (0 missing)  
## Surrogate splits:  
## Cell.shape splits as LLLRRRRRRR, agree=0.917, adj=0.8, (0 split)  
## Marg.adhesion splits as LLRRRRRRRR, agree=0.917, adj=0.8, (0 split)  
## Epith.c.size splits as LLLRRRRRRR, agree=0.875, adj=0.7, (0 split)  
## Bl.cromatin splits as LLLL--LR-R, agree=0.833, adj=0.6, (0 split)  
## Normal.nucleoli splits as LLR----LRR, agree=0.833, adj=0.6, (0 split)  
##   
## Node number 7: 143 observations  
## predicted class=malignant expected loss=0.06293706 P(node) =0.3496333  
## class counts: 9 134  
## probabilities: 0.063 0.937   
##   
## Node number 12: 14 observations  
## predicted class=benign expected loss=0 P(node) =0.03422983  
## class counts: 14 0  
## probabilities: 1.000 0.000   
##   
## Node number 13: 10 observations  
## predicted class=malignant expected loss=0.1 P(node) =0.02444988  
## class counts: 1 9  
## probabilities: 0.100 0.900

mytree.pred <- predict(mytree,eval.df,type="class")

ans <- numeric(length(eval.df[,1]))  
for (i in 1:length(eval.df[,1])) {  
 mytree <- rpart(Class ~ ., eval.df[-i,])  
 mytree.pred <- predict(mytree,eval.df[i,],type="class")  
 ans[i] <- mytree.pred  
}  
ans <- factor(ans,labels=levels(eval.df$Class))

myrda <- rda(Class ~ ., train.df)  
myrda.pred <- predict(myrda, eval.df)

myrf <- randomForest(Class ~ ., train.df)  
myrf.pred <- predict(myrf, eval.df)

#Convert predictions to df  
myrfResults <- data.frame(myrf.pred)  
myrdaResults <- data.frame(myrda.pred)  
LOOCVResults <- data.frame(ans)  
DTResults <- data.frame(mytree.pred)  
NNResults <- data.frame(mynnet.pred)  
NBResults <- data.frame(mynb.pred)  
SVMResults <- data.frame(mysvm.pred)  
  
#Create combinded df with all results  
combinedresults <- cbind(myrfResults[, 1], myrdaResults[, 1], LOOCVResults, DTResults, NNResults, NBResults[, 1], SVMResults)

## Warning in data.frame(..., check.names = FALSE): row names were found from a  
## short variable and have been discarded

#rename columns in combined df  
names(combinedresults) <- c("rfPred","rdaPred", "LOOCVPred", "DTPred", "NNPred", "NBPred","SVPred")  
  
#Convert results to numerical and sum  
sumPred <- ifelse(combinedresults$rfPred %in% "malignant",1,0) + ifelse(combinedresults$rdaPred %in% "malignant",1,0) + ifelse(combinedresults$LOOCVPred %in% "malignant",1,0) + ifelse(combinedresults$DTPred %in% "malignant",1,0)+ ifelse(combinedresults$NNPred %in% "malignant",1,0)+ ifelse(combinedresults$NBPred %in% "malignant",1,0)+ ifelse(combinedresults$SVPred %in% "malignant",1,0)  
  
#If majority malignant then malignant  
allPred <- ifelse(sumPred > 3, "malignant", "benign")

#create a df to store each models accuracy  
accuracy.df <- data.frame(PredModel = c("AllPred","rfPred","rdaPred", "LOOCVPred", "DTPred", "NNPred", "NBPred","SVPred"), accuracy = rep(0, 8))  
  
#Get AllPred Accuracy  
accuracy.df[1,2] <- confusionMatrix(as.factor(allPred), as.factor(eval.df$Class))$overall[1]  
  
#Get rfPred Accuracy  
accuracy.df[2,2] <- confusionMatrix(as.factor(combinedresults$rfPred), as.factor(eval.df$Class))$overall[1]  
  
#Get rdaPred Accuracy  
accuracy.df[3,2] <- confusionMatrix(as.factor(combinedresults$rdaPred), as.factor(eval.df$Class))$overall[1]  
  
#Get LOOCVPred Accuracy  
accuracy.df[4,2] <- confusionMatrix(as.factor(combinedresults$LOOCVPred), as.factor(eval.df$Class))$overall[1]  
  
#Get DTPred Accuracy  
accuracy.df[5,2] <- confusionMatrix(as.factor(combinedresults$DTPred), as.factor(eval.df$Class))$overall[1]  
  
#Get NNPred Accuracy  
accuracy.df[6,2] <- confusionMatrix(as.factor(combinedresults$NNPred), as.factor(eval.df$Class))$overall[1]  
  
#Get NBPred Accuracy  
accuracy.df[7,2] <- confusionMatrix(as.factor(combinedresults$NBPred), as.factor(eval.df$Class))$overall[1]  
  
#Get SVPred Accuracy  
accuracy.df[8,2] <- confusionMatrix(as.factor(combinedresults$SVPred), as.factor(eval.df$Class))$overall[1]  
  
#Return Accuracy Calculations  
accuracy.df

## PredModel accuracy  
## 1 AllPred 0.9744526  
## 2 rfPred 0.9708029  
## 3 rdaPred 0.9708029  
## 4 LOOCVPred 0.9306569  
## 5 DTPred 0.3321168  
## 6 NNPred 0.9489051  
## 7 NBPred 0.9744526  
## 8 SVPred 0.9708029

#Using the majority approach matched the best output from using any single model. Only 1 of the original models would have produced the same level of accuracy, but the other 6 if used in isolation would have produced a worse prediction. Using the ensemble corrects for the predictive deficiency of the other models.