Project Two

Benjamin Pope

3/8/2021

#Library  
library(e1071)

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

library(klaR)

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

library(nnet)  
library(neuralnet)

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

library(MASS)  
library(rpart)  
library(randomForest)

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

library(mlbench) #Includes BreastCancer data set.

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

library(caret)

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

library(stringr)

#Import data  
data("BreastCancer")  
  
# Set up columns correctly.   
# Set label as the first column in data frame.  
mydata <- cbind(BreastCancer[11],BreastCancer[2:10])

#Look at the data to see its structure and if there are any missing values.   
summary(mydata)

## Class Cl.thickness Cell.size Cell.shape Marg.adhesion  
## benign :458 1 :145 1 :384 1 :353 1 :407   
## malignant:241 5 :130 10 : 67 2 : 59 2 : 58   
## 3 :108 3 : 52 10 : 58 3 : 58   
## 4 : 80 2 : 45 3 : 56 10 : 55   
## 10 : 69 4 : 40 4 : 44 4 : 33   
## 2 : 50 5 : 30 5 : 34 8 : 25   
## (Other):117 (Other): 81 (Other): 95 (Other): 63   
## Epith.c.size Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses   
## 2 :386 1 :402 2 :166 1 :443 1 :579   
## 3 : 72 10 :132 3 :165 10 : 61 2 : 35   
## 4 : 48 2 : 30 1 :152 3 : 44 3 : 33   
## 1 : 47 5 : 30 7 : 73 2 : 36 10 : 14   
## 6 : 41 3 : 28 4 : 40 8 : 24 4 : 12   
## 5 : 39 (Other): 61 5 : 34 6 : 22 7 : 9   
## (Other): 66 NA's : 16 (Other): 69 (Other): 69 (Other): 17

str(mydata)

## 'data.frame': 699 obs. of 10 variables:  
## $ Class : Factor w/ 2 levels "benign","malignant": 1 1 1 1 1 2 1 1 1 1 ...  
## $ Cl.thickness : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 5 5 3 6 4 8 1 2 2 4 ...  
## $ Cell.size : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 1 1 2 ...  
## $ Cell.shape : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 4 1 8 1 10 1 2 1 1 ...  
## $ Marg.adhesion : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 1 5 1 1 3 8 1 1 1 1 ...  
## $ Epith.c.size : Ord.factor w/ 10 levels "1"<"2"<"3"<"4"<..: 2 7 2 3 2 7 2 2 2 2 ...  
## $ Bare.nuclei : Factor w/ 10 levels "1","2","3","4",..: 1 10 2 4 1 10 10 1 1 1 ...  
## $ Bl.cromatin : Factor w/ 10 levels "1","2","3","4",..: 3 3 3 3 3 9 3 3 1 2 ...  
## $ Normal.nucleoli: Factor w/ 10 levels "1","2","3","4",..: 1 2 1 7 1 7 1 1 1 1 ...  
## $ Mitoses : Factor w/ 9 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 5 1 ...

#Impute missing values with mean  
for (i in 1:ncol(mydata)) {  
 mydata[is.na(mydata[,i]), i] <- floor(mean(as.numeric(mydata[,i]), na.rm = TRUE))  
}  
mydata$Malignant\_1 <- ifelse(mydata$Class == "malignant",1,0) #Set our label to binary for our numerical data frame  
  
mydata.num <- as.data.frame(apply(mydata[,2:11],2,as.numeric))  
mydata.num <- cbind(mydata.num[10],mydata.num[1:9]) # Reorder our numerical data frame so that our label is first  
mydata <- mydata[,1:10] #Removes our numerical label from our factored data frame

#Partition Data  
t\_index <- sample(c(1:dim(mydata)[1]), dim(mydata)[1]\*.6) # Sets our training data to 60% of our data  
train.df <- mydata[t\_index, ]  
valid.df <- mydata[-t\_index, ]

#set up data frame for collecting accuracy  
accuracy.df <- data.frame(Model = seq(1, 8,1), Train\_Accuracy\_score = rep(0,8) ,Valid\_Accuracy\_score = rep(0,8))

# Support Vector machines  
accuracy.df[1,1] <- "Support Vector Machines"  
bdsvm <- svm(Class~.,train.df)  
bdsvm.pred <- predict(bdsvm,train.df)  
accuracy.df[1,2] <-confusionMatrix(as.factor(bdsvm.pred), as.factor(train.df$Class))$overall[1]  
#Create Predictions  
bdsvm.v.pred <- predict(bdsvm, valid.df)  
#put in accuracy to data frame  
accuracy.df[1,3] <- confusionMatrix(as.factor(bdsvm.v.pred), as.factor(valid.df$Class))$overall[1]

#Naive Bays  
accuracy.df[2,1] <- "Naive Bays"  
bdnvb <-NaiveBayes(Class ~., train.df)  
bdnvb.pred <- predict(bdnvb, train.df)

## 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 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 22

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 31

## 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 40

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 42

## 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 49

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 50

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 53

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 55

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 56

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 72

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 75

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 77

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 78

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 81

## 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 89

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 98

## 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 121

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 125

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 126

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 127

## 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 139

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 148

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 152

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 157

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 158

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 159

## 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 168

## 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 183

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 206

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 208

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 209

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 221

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 225

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 229

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 233

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 237

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 248

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 254

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 265

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 267

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 271

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 279

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 289

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 298

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 301

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 303

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 309

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 311

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 323

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 331

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 332

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 335

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 341

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 342

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 345

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 347

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 361

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 369

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 370

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 372

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 376

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 382

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 391

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 392

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 395

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 402

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 409

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 418

accuracy.df[2,2] <- confusionMatrix(as.factor(bdnvb.pred$class), as.factor(train.df$Class))$overall[1]  
#Create Predictions  
bdnvb.v.pred <- predict(bdnvb, valid.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 5

## 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 10

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 15

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 16

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 19

## 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 22

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 23

## 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 29

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 34

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 41

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 42

## 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 45

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 46

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 48

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 50

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 53

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 58

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 60

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 62

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 66

## 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 82

## 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 87

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 88

## 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 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 97

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 99

## 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 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 119

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 123

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 125

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 126

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 129

## 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 133

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 134

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 136

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 140

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 145

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 148

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 149

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 150

## 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 168

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 169

## 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 187

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 192

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 193

## 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 200

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 207

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 222

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 224

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 246

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 251

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 255

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 260

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 261

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 269

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 271

## Warning in FUN(X[[i]], ...): Numerical 0 probability for all classes with  
## observation 280

#Put in accuracy into data frame  
accuracy.df[2,3] <- confusionMatrix(as.factor(bdnvb.v.pred$class), as.factor(valid.df$Class))$overall[1]

#Neural Net  
#Set our training and validation datasets. \*NOTE be sure to use our numerical data frames for NN   
train.num.df <- mydata.num[t\_index, ]  
valid.num.df <- mydata.num[-t\_index, ]  
  
#normalize training and valid sets.  
norm\_values <- preProcess(train.num.df[,2:10])  
train.norm.df <- predict(norm\_values, train.num.df)  
valid.norm.df <- predict(norm\_values, valid.num.df)

#Create Neural Network  
accuracy.df[3,1] <- "Neural Net"  
bdnnet <- neuralnet(Malignant\_1 ~ .,linear.output = T, data = train.norm.df, hidden = c(2,5), rep = 5) #NN of 2 layers with 5 nodes.   
train.pred <- compute(bdnnet, train.norm.df)  
train.class <- ifelse(train.pred$net.result > .5, 1, 0)  
accuracy.df[3,2] <- confusionMatrix(as.factor(train.class), as.factor(train.num.df$Malignant\_1))$overall[1]  
#create Predictions  
valid.pred <- compute(bdnnet, valid.norm.df)  
valid.class <- ifelse(valid.pred$net.result > .5, 1, 0)  
#put in accuracy to data frame.  
accuracy.df[3,3] <- confusionMatrix(as.factor(valid.class), as.factor(valid.norm.df$Malignant\_1))$overall[1]

#Decision trees  
accuracy.df[4,1] <- "Decision Tree"  
bdtree <- rpart(Class~ ., train.df)  
#create predictions for DT  
bdtree.pred <- predict(bdtree, train.df, type = "class")  
bdtree.v.pred <- predict(bdtree, valid.df, type = "class")  
#put in accuracy to data frame.  
accuracy.df[4,2] <- confusionMatrix(as.factor(bdtree.pred), as.factor(train.df$Class))$overall[1]  
  
accuracy.df[4,3] <- confusionMatrix(bdtree.v.pred, valid.df$Class)$overall[1]

# Leave-1-Out Cross Validation (LOOCV)  
  
accuracy.df[5,1] <- "Leave-1-Out Cross Validation (LOOCV)"  
  
ans <- numeric(length(as.numeric(valid.df[,1])))  
for (i in 1:length(valid.df[,1])) {  
 bdtree2 <- rpart(Class ~ ., valid.df[-i,])  
 bdtree2.pred <- predict(bdtree, valid.df[i,],type="class")  
 ans[i] <- bdtree2.pred  
}  
ans <- factor(ans,labels=levels(valid.df$Class))  
  
accuracy.df[5,3] <- confusionMatrix(as.factor(ans), as.factor(valid.df$Class))$overall[1]  
# The same as above in this case

accuracy.df[6,1] <- "Regularized Discriminant Analysis"  
bdrda <- rda(Class ~ ., train.df)  
bdrda.pred <- predict(bdrda, train.df)  
accuracy.df[6,2] <- confusionMatrix(as.factor(bdrda.pred$class), as.factor(train.df$Class))$overall[1]  
  
bdrda2 <- rda(Class ~ ., mydata)  
bdrda.v.pred <- predict(bdrda, valid.df)  
accuracy.df[6,3] <- confusionMatrix(as.factor(bdrda.v.pred$class), as.factor(valid.df$Class))$overall[1]

#Random Forests  
accuracy.df[7,1] <- "Random Forests"  
bdrf <- randomForest(Class~., train.df, importance= TRUE)  
bdrf.pred <- predict(bdrf, train.df)  
accuracy.df[7,2] <- confusionMatrix(bdrf.pred, train.df$Class)$overall[1]  
# (Suspiciously correct! - need to read the manual)  
  
bdrf.v.pred <- predict(bdrf, valid.df)  
accuracy.df[7,3] <- confusionMatrix(as.factor(bdrf.v.pred), as.factor(valid.df$Class))$overall[1]

#Pull in results from three models  
  
  
ensamble.df <- cbind(as.data.frame(bdsvm.v.pred)[1],as.data.frame(bdnvb.v.pred)[1],as.data.frame(valid.class)[1],as.data.frame(bdtree.v.pred)[1],as.data.frame(ans),as.data.frame(bdrda.v.pred)[1],as.data.frame(bdrf.v.pred)[1])  
colnames(ensamble.df) <-c("svm", "nvb", "nnet","dectree","LOOCV", "rda", "rf" )  
  
ensamble.df$svm <- ifelse(ensamble.df$svm == "malignant",1,0)  
ensamble.df$nvb <- ifelse(ensamble.df$nvb == "malignant",1,0)  
ensamble.df$dectree <- ifelse(ensamble.df$dectree =="malignant",1,0)  
ensamble.df$rda <- ifelse(ensamble.df$rda == "malignant",1,0)  
ensamble.df$LOOCV <- ifelse(ensamble.df$LOOCV =="malignant",1,0)  
ensamble.df$rf <- ifelse(ensamble.df$rf =="malignant",1,0)  
#Use sum answer as output  
e <- as.matrix(ensamble.df) #matrix for rowsums  
#put our sums back onto the ensamble data frame.   
ensamble.df$combo <- rowSums(e)  
  
#Give us a class based answer again  
ensamble.df$combo\_class <- ifelse(ensamble.df$combo >=4,"malignant", "benign")  
#Check our accuracy with our ensamble score   
accuracy.df[8,1] <- "Combo Score "  
accuracy.df[8,3] <- confusionMatrix(as.factor(ensamble.df$combo\_class), as.factor(valid.df$Class))$overall[1]

#Here is all of our models accuaracies. We can take these results to our deployment when we decidee on which models to use.   
accuracy.df

## Model Train\_Accuracy\_score  
## 1 Support Vector Machines 0.9761337  
## 2 Naive Bays 0.9856802  
## 3 Neural Net 0.9952267  
## 4 Decision Tree 0.9785203  
## 5 Leave-1-Out Cross Validation (LOOCV) 0.0000000  
## 6 Regularized Discriminant Analysis 1.0000000  
## 7 Random Forests 1.0000000  
## 8 Combo Score 0.0000000  
## Valid\_Accuracy\_score  
## 1 0.9500000  
## 2 0.9535714  
## 3 0.9428571  
## 4 0.9107143  
## 5 0.9107143  
## 6 0.9464286  
## 7 0.9464286  
## 8 0.9500000