DATA 622 - Homework 2

OMER OZEREN

Table of Contents

## Load the Data

df <- read.table("C:/Users/OMERO/Documents/GitHub/DATA622/data.txt",header = T,sep=',')  
df$label <- ifelse(df$label =="BLACK",1,0)  
df$y <- as.numeric(df$y)  
df$X <- as.factor(df$X)

### Split Data into Train(70%) and Test data(30%)

set.seed(42)  
split\_df <- createDataPartition(df$label, p = .70, list = FALSE)  
df\_train <- df[split\_df,]  
df\_test <- df[-split\_df,]

nb.model<-naiveBayes(df\_train$label~.,data=df\_train)  
#str(nb.model)  
object.size(nb.model) #11096

## 5680 bytes

nb.tstpred<-predict(nb.model,df\_test[,-c(9)],type='raw')  
nb.tstclass<-unlist(apply(round(nb.tstpred),1,which.max))-1  
nb.tbl<-table(df\_test$label, nb.tstclass)  
nb.cfm<-caret::confusionMatrix(nb.tbl)  
nb.cfm

## Confusion Matrix and Statistics  
##   
## nb.tstclass  
## 0 1  
## 0 2 0  
## 1 2 6  
##   
## Accuracy : 0.8   
## 95% CI : (0.4439, 0.9748)  
## No Information Rate : 0.6   
## P-Value [Acc > NIR] : 0.1673   
##   
## Kappa : 0.5455   
##   
## Mcnemar's Test P-Value : 0.4795   
##   
## Sensitivity : 0.50   
## Specificity : 1.00   
## Pos Pred Value : 1.00   
## Neg Pred Value : 0.75   
## Prevalence : 0.40   
## Detection Rate : 0.20   
## Detection Prevalence : 0.20   
## Balanced Accuracy : 0.75   
##   
## 'Positive' Class : 0   
##

start\_tm <- proc.time()   
df<-df\_train  
runModel<-function(df) {naiveBayes(df$label~.,data=df[sample(1:nrow(df),nrow(df),replace=T),])}  
lapplyrunmodel<-function(x)runModel(df)  
system.time(models<-lapply(1:100,lapplyrunmodel))

## user system elapsed   
## 0.17 0.00 0.17

object.size(models)

## 568848 bytes

end\_tm<-proc.time()   
print(paste("time taken to run 100 bootstrapps",(end\_tm-start\_tm),sep=":"))

## [1] "time taken to run 100 bootstrapps:0.34"  
## [2] "time taken to run 100 bootstrapps:0"   
## [3] "time taken to run 100 bootstrapps:0.34"  
## [4] "time taken to run 100 bootstrapps:NA"   
## [5] "time taken to run 100 bootstrapps:NA"

bagging\_preds<-lapply(models,FUN=function(M,D=df\_test[,-c(9)])predict(M,D,type='raw'))  
bagging\_cfm<-lapply(bagging\_preds,FUN=function(P,A=df\_test$label)  
{pred\_class<-unlist(apply(round(P),1,which.max))-1  
 pred\_tbl<-table(A,pred\_class)  
 pred\_cfm<-caret::confusionMatrix(pred\_tbl)  
 pred\_cfm  
})  
bagging.perf<-as.data.frame(do.call('rbind',lapply(bagging\_cfm,FUN=function(cfm)c(cfm$overall,cfm$byClass))))  
bagging.perf.mean<-apply(bagging.perf[bagging.perf$AccuracyPValue<0.01,-c(6:7)],2,mean)  
bagging.perf.var<-apply(bagging.perf[bagging.perf$AccuracyPValue<0.01,-c(6:7)],2,sd)  
   
bagging.perf.var

## Accuracy Kappa AccuracyLower   
## NA NA NA   
## AccuracyUpper AccuracyNull Sensitivity   
## NA NA NA   
## Specificity Pos Pred Value Neg Pred Value   
## NA NA NA   
## Precision Recall F1   
## NA NA NA   
## Prevalence Detection Rate Detection Prevalence   
## NA NA NA   
## Balanced Accuracy   
## NA

bagging.perf.mean

## Accuracy Kappa AccuracyLower   
## NaN NaN NaN   
## AccuracyUpper AccuracyNull Sensitivity   
## NaN NaN NaN   
## Specificity Pos Pred Value Neg Pred Value   
## NaN NaN NaN   
## Precision Recall F1   
## NaN NaN NaN   
## Prevalence Detection Rate Detection Prevalence   
## NaN NaN NaN   
## Balanced Accuracy   
## NaN

(bagging\_tm<-proc.time()-start\_tm)

## user system elapsed   
## 0.75 0.09 0.82