stat460

Henrique Magalhaes Rio

4/18/2022

library(OpenImageR)  
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(ggbiplot)

## Loading required package: ggplot2

## Loading required package: plyr

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

## 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

## Loading required package: scales

## Loading required package: grid

library(dplyr)  
library(MASS)

##   
## Attaching package: 'MASS'

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

library(klaR)  
library(ggpubr)

##   
## Attaching package: 'ggpubr'

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

library(caret)

## Loading required package: lattice

library(class)

path="C:/Users/henri/econ436/Training/glioma\_tumor"  
  
setwd(path)  
Files <- list.files()  
Results <- list()  
for(i in seq\_along(Files)){  
 Image <- readImage(Files[i])  
 Resized <- resizeImage(Image, width = 200, height = 200)  
 Gray <- rgb\_2gray(Resized)  
 Results[[i]] <- Gray  
}

path="C:/Users/henri/econ436/Training/no\_tumor"  
  
setwd(path)  
Files <- list.files()  
Results2 <- list()  
for(i in seq\_along(Files)){  
 Image <- readImage(Files[i])  
 Resized <- resizeImage(Image, width = 200, height = 200)  
 Gray <- rgb\_2gray(Resized)  
 Results2[[i]] <- Gray  
}

## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !

## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !

## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !

path="C:/Users/henri/econ436/Testing/glioma\_tumor"  
  
setwd(path)  
Files <- list.files()  
Results3 <- list()  
for(i in seq\_along(Files)){  
 Image <- readImage(Files[i])  
 Resized <- resizeImage(Image, width = 200, height = 200)  
 Gray <- rgb\_2gray(Resized)  
 Results3[[i]] <- Gray  
}

## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !

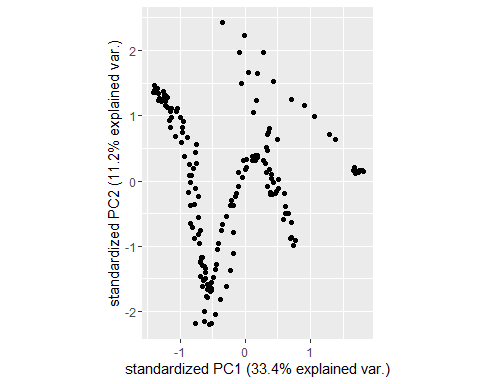
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !

## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the  
## 'height' becomes 199 therefore we have to adjust it to the input height  
## parameter of 200.000000 !

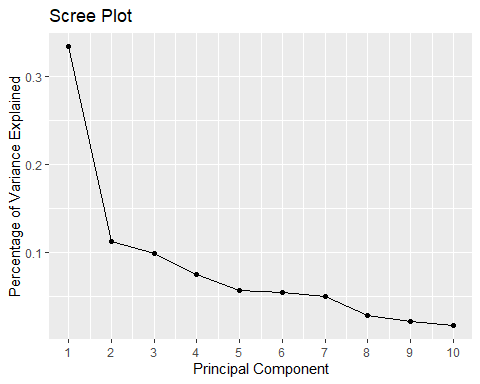
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !  
  
## Warning in resize\_nearest\_array(image, width, height): When resizing the 'width'  
## becomes 199 therefore we have to adjust it to the input width parameter of  
## 200.000000 !

path="C:/Users/henri/econ436/Testing/no\_tumor"  
  
setwd(path)  
Files <- list.files()  
Results4 <- list()  
for(i in seq\_along(Files)){  
 Image <- readImage(Files[i])  
 Resized <- resizeImage(Image, width = 200, height = 200)  
 Gray <- rgb\_2gray(Resized)  
 Results4[[i]] <- Gray  
}

pca<-prcomp(Results[[1]])  
  
compressed.img1 <- pca$x[,50:200] %\*% t(pca$rotation[,50:200])  
compressed.img2 <- pca$x[,1:50] %\*% t(pca$rotation[,1:50])  
ggbiplot(pca,var.axes = F)



var\_explained = pca$sdev^2 / sum(pca$sdev^2)  
  
ggplot()+geom\_line(aes(x=1:10,y=var\_explained[1:10]))+scale\_x\_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10))+geom\_point(aes(x=1:10,y=var\_explained[1:10]))+labs(x="Principal Component",y="Percentage of Variance Explained",title = "Scree Plot")



pca<-prcomp(Results2[[200]])  
  
compressed.img3 <- pca$x[,1:50] %\*% t(pca$rotation[,1:50])  
compressed.img4 <- pca$x[,50:200] %\*% t(pca$rotation[,50:200])

df <- data.frame(matrix(ncol = 200, nrow = 826))  
  
for(i in 1:826){  
  
pca<-prcomp(Results[[i]])  
row <-pca$x[,1]  
  
df[i,]<-row  
  
  
}

df2 <- data.frame(matrix(ncol = 200, nrow = 395))  
  
for(i in 1:395){  
  
pca<-prcomp(Results2[[i]])  
  
row <-pca$x[,1]  
  
df2[i,]<-row  
  
  
}

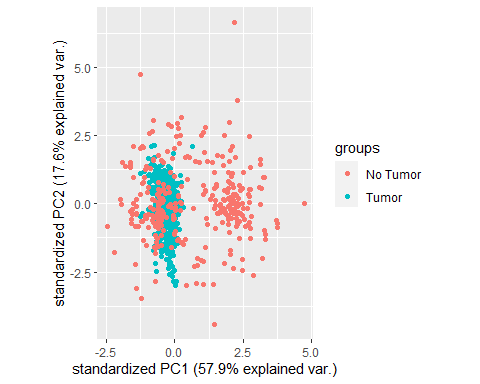
df3 <- data.frame(matrix(ncol = 200, nrow =100))  
  
for(i in 1:100){  
  
pca<-prcomp(Results3[[i]])  
  
row <-pca$x[,1]  
  
df3[i,]<-row  
  
  
}

df4 <- data.frame(matrix(ncol = 200, nrow =100))  
  
for(i in 1:100){  
  
pca<-prcomp(Results4[[i]])  
  
row <-pca$x[,1]  
  
df4[i,]<-row  
  
  
}

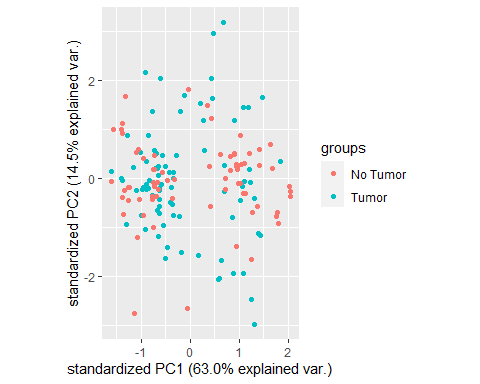
df<-df %>%  
 mutate(cancer="Tumor")  
  
df2 <-df2 %>% mutate(cancer="No Tumor")  
  
data <-rbind(df,df2)  
  
  
pca2 <- prcomp(data[,1:200],scale. = TRUE)

df3<-df3 %>%  
 mutate(cancer="Tumor")  
  
df4 <-df4 %>% mutate(cancer="No Tumor")  
  
testing <-rbind(df3,df4)  
  
  
pca3 <- prcomp(testing[,1:200],scale. = TRUE)

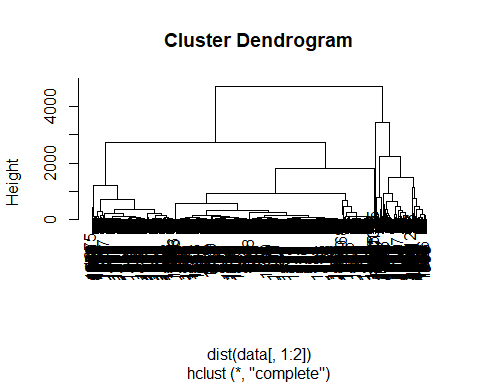
ggbiplot(pca2,groups = as.factor(data$cancer),var.axes = F)



ggbiplot(pca3,groups = as.factor(testing$cancer),var.axes = F)



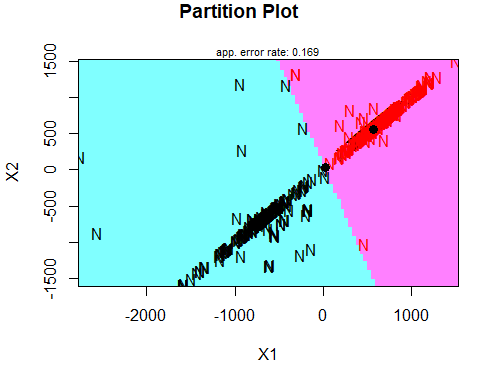
clust <- hclust(dist(data[,1:2]))  
  
plot(clust)



cutree <- cutree(clust,k=2)  
  
table(cutree)

## cutree  
## 1 2   
## 1035 186

lda <-partimat(as.factor(cancer) ~ X2+X1, data = data,method="lda")



logi <- glm(as.factor(cancer)~.,data=data,family="binomial")  
  
prob <- predict(logi, testing, type = "response")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from a rank-deficient fit may be misleading

pred <- ifelse(prob>0.5, "Tumor", "No Tumor")  
  
  
  
confusionMatrix(as.factor(pred),as.factor(testing$cancer))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Tumor Tumor  
## No Tumor 89 45  
## Tumor 11 55  
##   
## Accuracy : 0.72   
## 95% CI : (0.6523, 0.781)  
## No Information Rate : 0.5   
## P-Value [Acc > NIR] : 2.008e-10   
##   
## Kappa : 0.44   
##   
## Mcnemar's Test P-Value : 1.035e-05   
##   
## Sensitivity : 0.8900   
## Specificity : 0.5500   
## Pos Pred Value : 0.6642   
## Neg Pred Value : 0.8333   
## Prevalence : 0.5000   
## Detection Rate : 0.4450   
## Detection Prevalence : 0.6700   
## Balanced Accuracy : 0.7200   
##   
## 'Positive' Class : No Tumor   
##

lda2 <- lda(as.factor(cancer)~.,data=data)

## Warning in lda.default(x, grouping, ...): variables are collinear

predlda <- predict(lda2,newdata=testing)  
  
  
confusionMatrix(as.factor(predlda$class),as.factor(testing$cancer))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Tumor Tumor  
## No Tumor 76 41  
## Tumor 24 59  
##   
## Accuracy : 0.675   
## 95% CI : (0.6053, 0.7394)  
## No Information Rate : 0.5   
## P-Value [Acc > NIR] : 4.164e-07   
##   
## Kappa : 0.35   
##   
## Mcnemar's Test P-Value : 0.04719   
##   
## Sensitivity : 0.7600   
## Specificity : 0.5900   
## Pos Pred Value : 0.6496   
## Neg Pred Value : 0.7108   
## Prevalence : 0.5000   
## Detection Rate : 0.3800   
## Detection Prevalence : 0.5850   
## Balanced Accuracy : 0.6750   
##   
## 'Positive' Class : No Tumor   
##

knnModel <- train(as.factor(cancer) ~.,   
 data = data,   
 method = 'knn')

knn <-knn(train=data[,1:200],test=testing[,1:200],cl=data$cancer,k=1)  
  
confusionMatrix(as.factor(knn),as.factor(testing$cancer))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Tumor Tumor  
## No Tumor 100 66  
## Tumor 0 34  
##   
## Accuracy : 0.67   
## 95% CI : (0.6002, 0.7347)  
## No Information Rate : 0.5   
## P-Value [Acc > NIR] : 8.688e-07   
##   
## Kappa : 0.34   
##   
## Mcnemar's Test P-Value : 1.235e-15   
##   
## Sensitivity : 1.0000   
## Specificity : 0.3400   
## Pos Pred Value : 0.6024   
## Neg Pred Value : 1.0000   
## Prevalence : 0.5000   
## Detection Rate : 0.5000   
## Detection Prevalence : 0.8300   
## Balanced Accuracy : 0.6700   
##   
## 'Positive' Class : No Tumor   
##