Projeto

2024-04-14

if (!requireNamespace("plot.matrix", quietly = TRUE)) install.packages("plot.matrix")  
if (!requireNamespace("dbscan", quietly = TRUE)) install.packages("dbscan")  
if (!requireNamespace("ggplot2", quietly = TRUE)) install.packages("ggplot2")  
if (!requireNamespace("rpart", quietly = TRUE)) install.packages("rpart")  
if (!requireNamespace("rpart.plot", quietly = TRUE)) install.packages("rpart.plot")  
if (!requireNamespace("caret", quietly = TRUE)) install.packages("caret")  
  
library(plot.matrix)  
library(dbscan)  
library(ggplot2)  
library(rpart)  
library(rpart.plot)  
library(caret)  
library(dplyr)  
  
  
plotfigure <<- function(row,dataset)  
{  
 X = NULL  
 if(!is.null(nrow(dataset)))  
 {  
 X = data.frame(matrix(dataset[row,2:785],nrow=28))  
 }  
 else  
 {  
 X = data.frame(matrix(dataset[row,2:785],nrow=28))  
 }  
 m1 = data.matrix(X)  
 plot(m1, cex=0.5)  
}

# Introdução

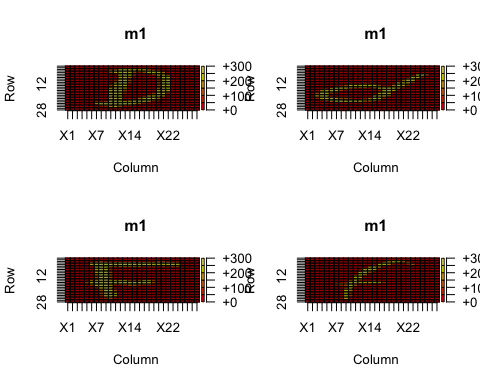
O presente trabalho tem como objetivo o desenvolvimento de um modelo que

A base de dados para o desenvolvimento do trabalho contém imagens de caracteres alfanuméricos manuscritos, incluindo letras maiúsculas e minúsculas, além de dígitos de 0 a 9. Cada imagem tem uma resolução de 28 × 28 = 784 pixeis. O trabalho será realizado para as letras D e F.

train\_data <<- read.csv("emnist-balanced-train.csv",sep=",",header = FALSE)  
test\_data <<- read.csv("emnist-balanced-test.csv",sep=",",header = FALSE)

Nas figuras seguintes pode observar-se algumas das possíveis representações da letra D e da letra F.

par(mfrow = c(2, 2))  
  
plotfigure(14,train\_data)  
plotfigure(60,train\_data)  
plotfigure(13,train\_data)  
plotfigure(4,train\_data)



Dado que se pretende apenas trabalhar com as letras D e F, reduziu-se a base de dados inicial para que esta apenas contenha as letras D e F.

label\_D <- 13  
label\_F <- 15  
  
filtered\_train\_data <- train\_data[train\_data$V1 %in% c(label\_D, label\_F), ]  
filtered\_test\_data <- test\_data[test\_data$V1 %in% c(label\_D, label\_F), ]  
  
head(filtered\_train\_data[, 1:5], 5)

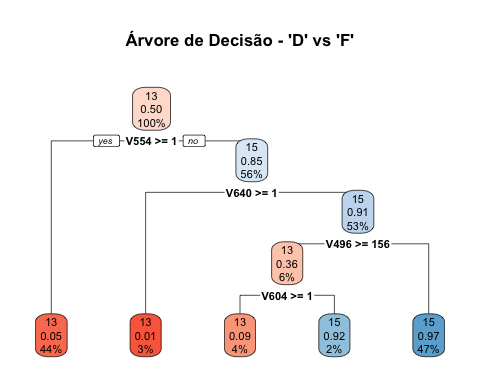
## V1 V2 V3 V4 V5  
## 4 15 0 0 0 0  
## 13 15 0 0 0 0  
## 14 13 0 0 0 0  
## 53 15 0 0 0 0  
## 73 13 0 0 0 0

# Análise Exploratória

Com o objetivo de distinguir as duas letras, é necessário determinar uma zona de relevância, ou seja, o conjunto de pixeis que determinam com elevada probabilidade se na imagem está representada a letra D ou a letra F.

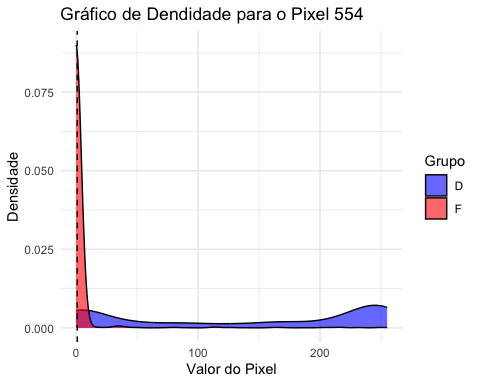
Com este intuito foi construída uma árvore de decisão, representada na figura seguinte.

set.seed(123)  
  
modelo\_arvore <- rpart(V1 ~ ., data = filtered\_train\_data, method = "class")  
  
rpart.plot(modelo\_arvore, main="Árvore de Decisão - 'D' vs 'F'",  
 box.palette = "RdBu")



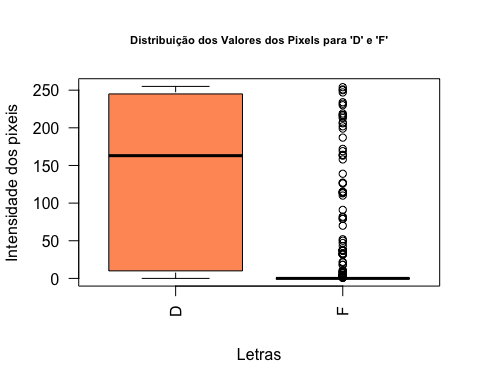
A árvore de decisão indica como pixel mais influente na diferenciação das letras o pixel 554. Se o valor do pixel for superior ou igual a 1, classifica-se a imagem como D. No entanto, se o pixel tiver um valor inferior a 1, o pixel 640 passa a ter influência para classificar a imagem. Isto significa que o pixel 554 é suficiente para classificar a imagem como D, mas não para classificar a imagem como F. A classificação da imagem como a letra F passa por analisar os pixeis 554, 640 e 496.

Seguidamente, estão representados os gráficos das densidades para os valores do pixel 554, para a letra D e para a letra F. A representação da letra D no gráfico é feita pela cor azul, enquanto que a letra F é exibida pela cor vermelha. A linha vertical a tracejado indica quando o valor do pixel é igual a 1. É visível que para imagens da letra F, existe uma predisposição maior para que o valores do pixel 554 sejam menores, enquanto que, para imagens da letra D, o pixel assume vários valores. (*comentar com árvore*)



Pela caixa-de-bigodes seguinte, é perceptível que para valores do pixel 554 inferiores a 1, existe uma distinção entre a letra D e F. Podemos observar que para as imagens da letra F, é mais comum o pixel 554 apresentar valores inferiores a 1 e para as imagens da letra D, é mais comum o pixel 554 apresentar valores superiores a 1.

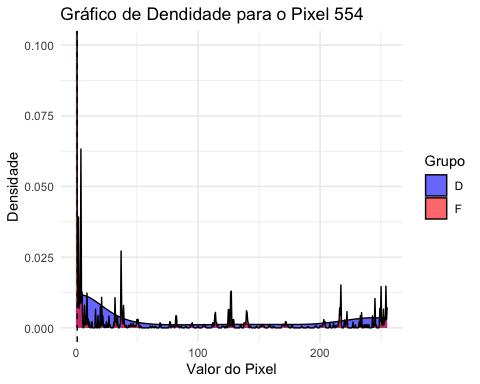
boxplot(as.matrix(filtered\_train\_data[,554:554]) ~ character\_labels,  
 col=c("#FF9966", "#996633"),   
 las=2,  
 xlab="Letras",  
 ylab="Intensidade dos pixeis",  
 #outline=FALSE  
)  
title(main = "Distribuição dos Valores dos Pixels para 'D' e 'F'", cex.main = 0.7)



par(mfrow = c(1, 2))  
  
ggplot(combined\_data, aes(x = V640, fill = group)) +  
 geom\_density(alpha = 0.6) +  
 scale\_fill\_manual(values = c("blue", "red")) +  
 theme\_minimal() +  
 labs(x = "Valor do Pixel", y = "Densidade", title = "Gráfico de Dendidade para o Pixel 554") +  
 guides(fill = guide\_legend(title = "Grupo")) +   
 geom\_vline(xintercept = 1, linetype = "dashed", color = "black") +  
 coord\_cartesian(ylim = c(0, 0.25))



#-----------------  
  
ggplot(combined\_data, aes(x = V496, fill = group)) +  
 geom\_density(alpha = 0.6) +  
 scale\_fill\_manual(values = c("blue", "red")) +  
 theme\_minimal() +  
 labs(x = "Valor do Pixel", y = "Densidade", title = "Gráfico de Dendidade para o Pixel 554") +  
 guides(fill = guide\_legend(title = "Grupo")) +   
 geom\_vline(xintercept = 1, linetype = "dashed", color = "black") +  
 coord\_cartesian(ylim = c(0, 0.1))



sums\_D <- colSums(filtered\_train\_data[filtered\_train\_data$V1 == label\_D, -1])  
sums\_F <- colSums(filtered\_train\_data[filtered\_train\_data$V1 == label\_F, -1])  
  
differences <- abs(sums\_D - sums\_F)  
  
  
matrix\_dim <- 28  
differences\_matrix <- matrix(differences, nrow = matrix\_dim, ncol = matrix\_dim)  
  
  
image(1:28, 1:28, differences\_matrix,  
 col = cm.colors(20),  
 xlab = "Pixeis", ylab = "Pixeis",  
 main = "Diferença dos Valores dos Pixeis")

