Universidade Estadual de Campinas - Instituto de Computação (IC/Unicamp)
Disciplina Aprendizado de Máquina (MO444)
Professor: Jacques Wainer **Aluno: Luiz Alberto Ferreira Gomes**
Atividade 1, 26 de novembro de 2018 **RA:007275**

Questão 1: [0 points]

Faça o PCA dos dados (sem a última coluna). Se voce quiser que os dados transformados tenham 80% da variância original, quantas dimensões do PCA vc precisa manter?

Solução:

A Tabela 1 apresenta a saída parcial do comando `summary(pca.output)`. A variável `pca.output` é a variável do script em R (ver Anexo I) que armazena o retorno da chamada da função `prcomp` que, por sua vez, calcula o PCA sobre os dados originais. Nessa tabela, a soma acumulada da variância (*Cumulative Proportion*) é maior do que 80% somente a partir do componente 13 (PCA13). Então, para se atingir a variância pedida no exercício serão necessárias 13 dimensões.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13
Standard deviation	7.1953	4.8073	3.5561	2.9219	2.8577	2.6019	2.3214	2.2474	1.8183	1.6883	1.6036	1.5417	1.4858
Proportion of Variance	0.3119	0.1392	0.0762	0.0514	0.0492	0.0408	0.0325	0.0304	0.0199	0.0172	0.0155	0.0143	0.0133
Cumulative Proportion	0.3119	0.4511	0.5273	0.5787	0.6279	0.6687	0.7012	0.7316	0.7515	0.7687	0.7842	0.7985	0.8118

Tabela 1: Saída do comando summary da linguagem R

Questão 2: [0 points]

Treine uma regressão logística no conjunto de treino dos dados originais e nos dados transformados. Qual a taxa de acerto no conjunto de teste nas 2 condições (sem e com PCA)?

Solução:

A taxa de acertos (ou acurácia) pode ser calculada a partir dos dados extraídos das tabelas de confusão, geradas pelo script em R (ver Anexo I), para as duas condições pedidas no exercício:

GLM sem PCA:

		Predição	
		0	1
Real	0	TN = 115	FP = 56
	1	FN = 44	TP = 61

Tabela 2: Tabela de confusão.

GLM com PCA:

		Predição	
		0	1
Real	0	TN = 73	FP = 52
	1	FN = 86	TP = 65

Tabela 3: Tabela de confusão.

$$\begin{aligned} Acuracia &= \frac{TN+TP}{TN+TP+FN+FP} \\ &= \frac{115+61}{115+61+44+56} \\ &= 0.6376 \end{aligned} \quad (1)$$

$$\begin{aligned} Acuracia &= \frac{TN+TP}{TN+TP+FN+FP} \\ &= \frac{73+65}{73+65+86+52} \\ &= 0.5000 \end{aligned} \quad (2)$$

A taxa de acerto do GLM sem pca foi de **0.6376** (equação 1) e do GLM com pca foi **0.5000** (equação 2).

Universidade Estadual de Campinas - Instituto de Computação (IC/Unicamp)
Disciplina Aprendizado de Máquina (MO444)
Professor: Jacques Wainer **Aluno: Luiz Alberto Ferreira Gomes**
Atividade 1, 26 de novembro de 2018 **RA:007275**

Questão 3: [0 points]

Treine o LDA nos conjuntos de treino com e sem PCA e teste nos respectivos conjuntos de testes. Qual a acurácia nas 2 condições?

Solução: A taxa de acertos (ou acurácia) pode ser calculada a partir dos dados extraídos das tabelas de confusão, geradas pelo script em R (ver Anexo I), para as duas condições pedidas no exercício:

LDA sem PCA:

		Predição	
		0	1
Real	0	TN = 127	FP = 48
	1	FN = 32	TP = 69

Tabela 4: Tabela de confusão.

LDA com PCA:

		Predição	
		0	1
Real	0	TN = 73	FP = 52
	1	FN = 86	TP = 65

Tabela 5: Tabela de confusão.

$$\begin{aligned} Acuracia &= \frac{TN+TP}{TN+TP+FN+FP} \\ &= \frac{127+69}{127+69+32+48} \\ &= 0.7101 \end{aligned} \quad (3)$$

$$\begin{aligned} Acuracia &= \frac{TN+TP}{TN+TP+FN+FP} \\ &= \frac{73+65}{73+65+86+52} \\ &= 0.5000 \end{aligned} \quad (4)$$

A taxa de acerto do LDA sem pca foi de **0.7101** (equação 3) e do LDA com pca foi **0.5000** (equação 4).

Questão 4: [0 points]

Qual a melhor combinação de classificador e PCA ou não?

Solução: A Figura 1, com as acurácias geradas pelo script em R (ver Anexo I), mostra que a melhor combinação de classificadores e PCA - responsável pela maior precisão - foi **o LDA sem PCA.**

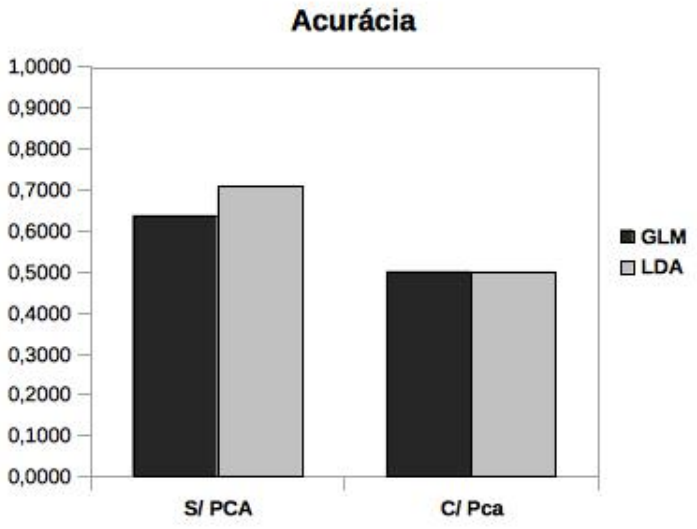

Figura 1: Gráfico das acurácias dos classificadores.

```
# ------------------------------------------------------------------------------
# Description:
#       Atividade 1 da disciplina MO444
#
# Version: 1.0
#
# Author:
#               Luiz Alberto, gomes.luiz@gmail.com
#
# History:
#   Sep 12th,   2016  atividade 1  started
#   Sep 13th,   2016  atividade 1  updated
#   Sep 14th,   2016  atividade 1  updated
#   Sep 17th,   2016  atividade 1  conclued.
#
# To do:
#  -
# ------------------------------------------------------------------------------

# installs the required packages.
if (!require("MASS"))
  install.packages("MASS")

require(MASS)

# sets the working directory
setwd('/Workspace/doutorado/disciplinas/mo444b/atividades/1')

# common functions
CalculateAccuracy <- function(confusion.matrix) {
  # Computes the accuracy value based on the confusion matrix
  #
  # Args:
  #   confusion.matrix: the confusion matrix with the values of predicting.
  #
  # Returns:
  #   The accuracy value.
  TN <- confusion.matrix[1, 1]
  FN <- confusion.matrix[2, 1]
  FP <- confusion.matrix[1, 2]
  TP <- confusion.matrix[2, 2]
  accuracy <- (TP + TN) / (TN + FN + FP + TP)
  return(accuracy)
}
#-------------------------------------------------------------------------------
# question 1
#-------------------------------------------------------------------------------
original.data <- read.csv(file = './data/data1.csv', header = TRUE, sep = ',')

# the original data minus the last column.
pca.input <- original.data[, 1:ncol(original.data) - 1]
```

```
# applies the pca method using prcomp function.
pca.output <- prcomp(pca.input, scale. = T)

# prints de pca output for analysis.
summary(pca.output)

# generates the original data using 13 pca components (according to previous
# analysis).
transformed.data <- as.data.frame(pca.output$x[, 1:13] %*%
                                  t(pca.output$rotation[, 1:13]))

# restores the last to column to data transformed set.
transformed.data$clase <- original.data[, 167]

#-------------------------------------------------------------------------------
# question 2
#-------------------------------------------------------------------------------

#
# GLM without PCA
#

# defines the formula for all classifiers.
classifier.formula = clase ~ f1 + f2 + f3 + f4 + f5 + f6 + f7 + f8 + f9 + f10 +
      f11 + f12 + f13

# separates the data set in train and test data
original.data.train <- original.data[1:200,]
original.data.test <- original.data[201:476, ]

# defines preditive column
preditive.column <- original.data$clase[201:476]

# trains the glm classifier.
glm.fit   <- glm(classifier.formula, data = original.data.train,
                 family = binomial(link = "logit"))

# tests predicties with data test.
glm.probs      <- predict.glm(glm.fit, original.data.test, type = "response")
glm.preditive  <- rep(0, nrow(original.data.test))
glm.preditive[glm.probs > 0.5] <- 1

# prints de confusion matrix
print('Confusion matrix for GLM without PCA')
print(table(glm.preditive, preditive.column), auto = TRUE)

# calculates an prints accuracy
print('Accuracy for GLM without PCA')
accuracy <- CalculateAccuracy(table(glm.preditive, preditive.column))
print(accuracy)

#
```

```
# GLM with PCA
#

# reads train data from transformed data by pca.
transformed.data.train <- transformed.data[1:200, ]

# trains the glm classifier.
glm.fit <- glm(classifier.formula, data = transformed.data.train,
      family = binomial(link = "logit"))

# tests predicties with data test.
glm.probs <- predict.glm(glm.fit, original.data.test, type = "response")
glm.preditive  <- rep(0, nrow(original.data.test))
glm.preditive[glm.probs > 0.5] = 1

# prints de confusion matrix
print('Confusion matrix for GLM with PCA')
print(table(glm.preditive, preditive.column), auto = TRUE)

# calculates an prints accuracy
print('Accuracy for GLM with PCA')
accuracy <- CalculateAccuracy(table(glm.preditive, preditive.column))
print(accuracy)

#-------------------------------------------------------------------------------
# question 3
#-------------------------------------------------------------------------------

#
# LDA without PCA
#

# trains the lda classifier.
lda.fit <- lda(classifier.formula, data = original.data,
               subset = original.data.train$x)

# tests predicties with data test.
lda.preditive  <- predict(lda.fit, original.data.test)
lda.class      <- lda.preditive$class

# prints de confusion matrix
confusion.matrix <- table(lda.class, preditive.column)
print('Confusion matrix for LDA without PCA')
print(confusion.matrix, auto = TRUE)

# calculates an prints accuracy
print('Accuracy for LDA without PCA')
accuracy <- CalculateAccuracy(confusion.matrix)
print(accuracy)

#
# LDA with PCA
```

```
#

# trains the lda classifier with transformed data by pca.
lda.fit <- lda(classifier.formula, data = transformed.data,
    subset = transformed.data.train$x)

# tests predicties with data test.
lda.preditive  <- predict(lda.fit, original.data.test)
lda.class      <- lda.preditive$class

# prints de confusion matrix
confusion.matrix <- table(lda.class, preditive.column)
print('Confusion matrix for LDA with PCA')
print(confusion.matrix, auto = TRUE)

# calculates an prints accuracy
print('Accuracy for LDA with PCA')
accuracy <- CalculateAccuracy(confusion.matrix)
print(accuracy)
```