Nome: Nathana Facion

Exercício 1

1. faca 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?
Gere os dados transformados mantendo 80\% da variância. (Atenção este passo não é 100\% correto do ponto de vista de aprendizado de maquina. Não repita este passo em outras atividades).
Considere as primeiras 200 linhas dos dados como o conjunto de treino, e as 276 ultimas como o conjunto de dados.
Resposta: 13 dimensões

2. 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)?
Glm_acuracia_with_pca = 0.7934783 (GLM e PCA)
glm_acuracia_origin= 0.6521739 (GLM e sem PCA)

3. Treine o LDA nos conjuntos de treino com e sem PCA e teste nos respectivos conjuntos de testes. Qual a acurácia nas 2 condições?
Lda_acuracia_with_pca = 0.7862319 (LDA e PCA)
Lda_acuracia_origin = 0.6775362 (LDA e sem PCA)

4. Qual a melhor combinação de classificador e PCA ou não?
GLM e PCA.

Saída:

```
[1] "Dimensao:"
[1] 13
[1] "glm_acuracia_with_pca:"
[1] 0.7934783
[1] "glm_acuracia_origin:"
[1] 0.6521739
[1] "lda_acuracia_with_pca:"
[1] 0.7862319
[1] "lda_acuracia_origin:"
[1] 0.6775362
```

Código:

```
# leitura do arquivo
all_data <- read.csv("//home//nathana//AM//data1.csv")

# numero de colunas total
number_column <- ncol(all_data)

# numero de linhas total
number_row <- nrow(all_data)

# selecionando ultima coluna que contem classe
class <- all_data[number_column]

# dados originais sem ultima coluna
data_origin <- all_data[1: number_column - 1]

# numero de coluna sem classe
number_col_without_class <- ncol(data_origin)

# usando a funcao de pca
pca <- prcomp(data_origin, scale. = T)

# fazendo a variancia para ver a quantidade de dimensao selecionada
sum_pca <- cumsum(pca$sdev^2)/sum(pca$sdev^2)

# porcentagem de variancia que deve ser aceita
variance_ini <- 0.8

# verifica qual eh a posicao desta variancia
dimensao = 0;
for (i in 1:(number_col_without_class)) {
  if (sum_pca[i] >= variance_ini){
          dimensao <- i
          break
  }
}


# gerando os dados transformados com as dimensoes da variancia
```

```
data_with_pca <-pca$x[,1:dimensao]

# numbero de dimensao
total_dimensions <- ncol(data_with_pca)
print("Dimensao:")
print(total_dimensions)

######### testes ############

# classes
class_treino <- class[c(1:200),1]
class_teste <- class[c(201:number_row),1]


# Selecionando treino e teste com dados originais
data_origin_treino <- data_origin[c(1:200), 1:number_col_without_class]
data_origin_treino["class"] <- data.frame(class_treino)
data_origin_teste <- data_origin[c(201:number_row), 1:number_col_without_class]
data_origin_teste["class"] <- data.frame(class_teste)

# Selecionando treino com dados com PCA
data_with_pca_treino <- data_with_pca[c(1:200), 1:ncol(data_with_pca)]
data_with_pca_treino <- data.frame(data_with_pca_treino)
data_with_pca_treino["class"] <- data.frame(class_treino)
data_with_pca_teste <- data_with_pca[c(201:number_row), 1:ncol(data_with_pca)]
data_with_pca_teste <- data.frame(data_with_pca_teste)
data_with_pca_teste["class"] <- data.frame(class_teste)

# aplicando LDA e GLM
library(MASS)

### Com PCA ###
# Regressao logistica
data_with_pca_glm <- glm(formula = class ~ . , data = data.frame(data_with_pca_treino),
family=binomial(link=logit))
pred_with_pca_glm <-
predict(data_with_pca_glm,newdata=data.frame(data_with_pca_teste),type="response")

#LDA
data_with_pca_lda <- lda( class ~ .,data_with_pca_treino)
pred_with_pca_lda <- predict(data_with_pca_lda,data_with_pca_teste)
```

```
### Sem PCA ###
# Regressao logistica
data_origin_glm <- glm(formula = class ~ ., data =
data.frame(data_origin_treino),family=binomial(link=logit))
pred_origin_glm <-
predict(data_origin_glm,newdata=data.frame(data_origin_teste),type="response")

#LDA
data_origin_lda <- lda(class ~ .,data_origin_treino)
pred_origin_lda <- predict(data_origin_lda,data_origin_teste)


# Calculo de acuracia
# Questao  numero 2 -Treine uma regressao logistica no conjunto de treino dos dados originais
# e nos dados transformados.
# Qual a taxa de acerto no conjunto de teste nas 2 condicoes (sem e com PCA)
# Com PCA
glm_confusion_with_pca <- table(class_teste, pred_with_pca_glm > 0.5)
glm_acuracia_with_pca <- (glm_confusion_with_pca[1] +
glm_confusion_with_pca[4])/(glm_confusion_with_pca[1] + glm_confusion_with_pca[2] +
glm_confusion_with_pca[3] + glm_confusion_with_pca[4])
print("glm_acuracia_with_pca:")
print(glm_acuracia_with_pca)
# Sem PCA
glm_confusion_origin <- table(class_teste, pred_origin_glm > 0.5)
glm_acuracia_origin <- (glm_confusion_origin[1] +
glm_confusion_origin[4])/(glm_confusion_origin[1] + glm_confusion_origin[2] +
glm_confusion_origin[3] + glm_confusion_origin[4])
print("glm_acuracia_origin:")
print(glm_acuracia_origin)

# Questao numero 3 - Treine o LDA nos conjuntos de treino com e sem PCA e teste nos
#respectivos conjuntos de testes. Qual acuracia de cada um?
# Com PCA
lda_confusion_with_pca <- table(pred_with_pca_lda$class, class_teste)
lda_acuracia_with_pca <-(lda_confusion_with_pca[1] +
lda_confusion_with_pca[4])/(lda_confusion_with_pca[1] + lda_confusion_with_pca[2] +
lda_confusion_with_pca[3] + lda_confusion_with_pca[4])
print("lda_acuracia_with_pca:")
print(lda_acuracia_with_pca)
# Sem PCA
lda_confusion_origin <- table(pred_origin_lda$class, class_teste)
```

```
lda_acuracia_origin <-(lda_confusion_origin[1] +
lda_confusion_origin[4])/(lda_confusion_origin[1] + lda_confusion_origin[2] +
lda_confusion_origin[3] + lda_confusion_origin[4])
print("lda_acuracia_origin:")
print(lda_acuracia_origin)
```