```
######################### Esame 28/01/2019 ################################
#
###########################################################################



# Obiettivo prevede la specie del fiore utilzzando le variabili del dataset



####### 1 #######

#### Caricare le librerie
library(MASS)
library(ggplot2)
library(class) # knn
library(boot)
library(leaps)
library(outliers)
#### Caricare il dataset
mydf <- iris


# 1.1 ANALISI ESPLORATIVA
# Visualizzo il data set e faccio una ANALISI ESPLORATIVA con le funzioni base
# noto che il data set ha una dimensione ridotta
# ci sono 80 osservazioni e 6 predittori + una variabile di risposta (Caesarian)
# Le osservazioni riguardano 80 parti di donne, in cui si analizzano alcuni dati sanitari fondamentali
# e probabilmente correlabili alla fine di valutare la necessit? di un taglio cesario
# Osservo che la variabile Caesarian


#### Verifica generale del Dataset
summary(mydf)
dim(mydf)

# Il Dataset Ã¨ compostoda da 150 osservazioni e 5 variabili
# La variabile di risposta Ã¨ denominato
#             Species: variabile categorica che indica la specie di pianta suddivisa in tre classi equiripartite
#                   - setosa
#                   - versicolor
#                   - virginica
# Il Dataset Ã¨ composto da ulteriori 4 variabili predittori riguardanti le caratteristiche delle piante
#             Sepal.Width
#             Petal.Width
#             Sepal.Length
#             Petal.Length


# Verifico le prime osservazioni del Dataset
head(mydf)

# VErifico le ultime osservazioni del Dataset
tail(mydf)

# Verifico la struttura e classi del Dataset
str(mydf)


# 1.2 DATA CLEANING e DATA TRANSFORMATION


### Verifico eventuali NA
sum(is.na(mydf))
# Il dataset non presenta valori NA

### 1.3 RICERCA DI OUTLIERS O HIGH LEVERAGE POINTS
# Verifica della presenza di valori che non seguono l'andamento generale del data set
# pertanto sono da analizzare con estrema attenzione e valutare caso per caso il da farsi
# la ricerca e' supportata da grafici

summary(mydf)

# Analizzo Age, reputo altamente significativa la distribuzione di frequenza dell'istogramma
# sono dati ospedaliari, sono stati presi per una ricerca? Sembrano distribuiti
hist(mydf$Sepal.Length)
hist(mydf$Sepal.Width)
hist(mydf$Petal.Length)
hist(mydf$Petal.Width)

# calcoliamo gli outlier
# non trovo significativite motivazioni per eliminare id 4
mydf[outlier( mydf$Sepal.Length, opposite = FALSE, logical = FALSE),]
```

```
#La fuznione riestituisce come outlier la seguente osservazione.
# SarÃ  cura investigarla nel proseguo della trattazione
# Sepal.Length Sepal.Width Petal.Length Petal.Width Species
#       4          4.6         3.1          1.5         0.2  setosa

# Esperimento elimino il 4
mydf = mydf[-4,]

# ... Oppure: Non rilevo particolari osservazioni che potrebbero influenza l'analisi. Non rimuovo nessun valore.


# Verifico con una boxplot la presenza degli outlier.
boxplot(mydf$Sepal.Length)
boxplot(mydf$Sepal.Width)
boxplot(mydf$Petal.Length)
boxplot(mydf$Petal.Width)

pairs(mydf)



### 1.4 ANALISI GRAFICA DELLE RELAZIONI TRA LE VARIABILI

#GRAFICO26
# Correlazioni
# La funzione non puo' avere NA
corrplot::corrplot(cor(mydf[,sapply(mydf, is.numeric)]),
                   method = "number", type = "upper", order = "AOE",
                   title = "Matrice di correlazione lineare")

# Dal corrplot emerogno delle relazioni lineari molto forti tra:
#   Sepal.Lenght ->> con Petal.Width e Petal.Lenght
#   Petal.Width  ->> con Petal.Lenght. Questa Ã¨ una correlazione fortissima pari a 0.96 (quasi un rapporto 1:1 nella
lineare)
#   Petal.Lenght ->> con Sepal.Width


#GRAFICO27
# Analiziamo con pairs il rapporco con tutte le variabili a coppie
pairs(mydf, main="Matrice degli scatterplot", col="blue")

# La verifica degli scatterplot conferma la forte relazione:
#    Petal.Width  ->> con Petal.Lenght

#GRAFICO28
# DENSITA' per Sepal
ggplot2::ggplot(data = mydf) +
  geom_density(  mapping= aes(x=mydf$Sepal.Length), alpha=0.3, fill="Red") + xlab("Varie caratteristiche Sepal") +
ylab("Densita'") + ggtitle("Grafico di densita' per le caratteristiche della pianta") +
  geom_density(  mapping= aes(x=mydf$Sepal.Width), alpha=0.3, fill="green") + xlab("Varie caratteristiche Sepal") +
ylab("Densita'") + ggtitle("Grafico di densita' per le caratteristiche della pianta")

# DENSITA' per Petal
ggplot2::ggplot(data = mydf) +
  geom_density(  mapping= aes(x=mydf$Petal.Length), alpha=0.3, fill="blue") + xlab("Varie caratteristiche Petal ") +
ylab("Densita'") + ggtitle("Grafico di densita' per le caratteristiche della pianta") +
  geom_density(  mapping= aes(x=mydf$Petal.Width), alpha=0.3, fill="black") + xlab("Varie caratteristiche Petal") +
ylab("Densita'") + ggtitle("Grafico di densita' per le caratteristiche della pianta")

#GRAFICO29
boxplot(Sepal.Length~Species, data = mydf,
        horizontal=TRUE,
        xlab="Lunghezza del Sepalo",
        col=c("thistle","wheat"),
        main="Analisi per la lunghezza del sepalo")



# BUBBLE - BOLLE - Sepal.Lenght(continui) - Species (categorica)
# Il grafico a bolle mi da una rappresentazione relativa alla carratteristica osservata.
dt <- as.data.frame( aggregate(x=mydf$Sepal.Length, by=list(Age=mydf$Species), FUN=mean))
ggplot(dt, aes(x=dt$Age, y=dt$x, size=dt$x)) +
  geom_point(alpha=4/10) +
  xlab("le tre specie coinvolte") + ylab("Dimensioni") +
  ggtitle("Rapporto di dimensioni suddiviso per specie")

# BUBBLE - BOLLE - Petal.Width(continui) - Species (categorica)
# Il grafico a bolle mi da una rappresentazione relativa alla carratteristica osservata.
dt <- as.data.frame( aggregate(x=mydf$Petal.Width, by=list(Age=mydf$Species), FUN=mean))
ggplot(dt, aes(x=dt$Age, y=dt$x, size=dt$x)) +
  geom_point(alpha=4/10) +
  xlab("le tre specie coinvolte") + ylab("Dimensioni") +
  ggtitle("Rapporto di dimensioni suddiviso per specie")
```

```
####### 2 #######
#
# Sviluppare adeguati  modelli previsionali, utilizzando come variabile di risposta Caesarian. Commentare
# i modelli che si e' sviluppato, motivando quello ritenuto migliore (considerando sia l'aspetto predittivo,
# sia quello inferenziale)
#


# Si tratta anzitutto di un problema di classificazione. Infatti Species e' una variabile qualitativa
# che puo' assumere tre valori (SETOSA - VERSICOLOR - VIRGINICA).


# Dall'esame 1.4 ho potuto notare forti relazioni tra le variabili a livello lineare.
# Come modalitÃ  di verifica si propongono dei modelli di classificazione tipo LDA.
# Passi LDA:
#   (i) Si assume che la distribuzione delle X in ogni classe Ã¨ distribuita Normalmente ( o approssimativamente come
una Normale)
#   (ii) Si stimano le medie delle X in ciascuna classe, la varianza comune e le proporzioni di casi per ciascuna
classe.
#   (iii) Si applica il teorema di Bayes per calcolare la probabilitÃ  pk(x) e si assegna un caso alla classe con
pk(x) maggiore.
#   (iv) Si calcola la qualitÃ  della classificazione (o il tasso di errata classificazione)

### TODO il punto (i) Ã¨ intuitivo in quanto si tratta di un probelma naturale, per cui ci aspettiamo che risponda
alla normale


#### Validation approch
# Per realizzare il modello distinguo tra training set e test set
# con il data set preparo il modello
# con il test set ne valuto l'accuratezza
set.seed(1)
nperc = 0.8
index_train = sample(1:nrow(mydf), nperc * nrow(mydf), replace = F)


###### 2.1 LDA

# Provo a verificare con la linear discriminant analysis
#### Poche osservazioni LDA
#### Molte osservazione QDA

mod_lda <- lda(Species ~ Petal.Length + Sepal.Length, data = mydf, subset = index_train )

# Verifico il modello
mod_lda
summary(mod_lda)
plot(mod_lda)


# Analizzo le prioritÃ  a priori
# Prior probabilities of groups:
#  setosa versicolor  virginica
# 0.3333333  0.3500000  0.3166667


# faccio la predizione sul test set e valuto l'accuratezza
pred_lda = predict(mod_lda, mydf[-index_train,] )

# i risultati che ottengo si avvicinano molto a quelli ottenuti con la logistica
table(pred_lda$class, mydf[-index_train,  which(colnames(mydf)=="Species")])

mean(pred_lda$class == mydf[-index_train, which(colnames(mydf)=="Species")])
(1 - mean(pred_lda$class ==  mydf[-index_train,  which(colnames(mydf)=="Species")] )) #tasso  di err. clas.


###### 2.2 LDA  Rappresentazione grafica #####
#GRAFICO30
ggplotLDAPrep <- function(x){
  if (!is.null(Terms <- x$terms)) {
    data <- model.frame(x)
    X <- model.matrix(delete.response(Terms), data)
    g <- model.response(data)
    xint <- match("(Intercept)", colnames(X), nomatch = 0L)
    if (xint > 0L)
      X <- X[, -xint, drop = FALSE]
  }
  means <- colMeans(x$means)
  X <- scale(X, center = means, scale = FALSE) %*% x$scaling
  rtrn <- as.data.frame(cbind(X,labels=as.character(g)))
  rtrn <- data.frame(X,labels=as.character(g))
  return(rtrn)
}

test<-iris[grep("setosa|versicolor|virginica", iris$Species),1:5]
```

```
fitGraph <- ggplotLDAPrep(mod_lda)
ggplot(fitGraph, aes(LD1,LD2, color=labels))+
  geom_point() +
  stat_ellipse(aes(x=LD1, y=LD2, fill = labels), alpha = 0.2, geom = "polygon")


###### 2.2 KNN  Rappresentazione grafica #####


# i due precedenti modelli non si sono distinti significativamente
# verifichiamo con KNN

train_knn = cbind(mydf$Petal.Length[index_train], mydf$Sepal.Length[index_train], mydf$Sepal.Width[index_train],
mydf$Petal.Width[index_train])
test_knn = cbind(mydf$Petal.Length[-index_train], mydf$Sepal.Length[-index_train], mydf$Sepal.Width[-index_train],
mydf$Petal.Width[-index_train])

knn_pred = knn(train_knn, test_knn, mydf$Species[index_train], k = 11)

# Verifica accuratezza del modello
mean(knn_pred == mydf$Species[-index_train])

# Matrice di confusione
table(knn_pred, mydf$Species[-index_train])


#GRAFICO31
### 2.3 LOOCV ricerca del miglior K in KNN
set.seed(123)
train_tot_knn = cbind(mydf$Petal.Length, mydf$Sepal.Length, mydf$Sepal.Width, mydf$Petal.Width)
loocv.err = rep(0, nrow(mydf))
for(i in 1: (nrow(mydf)-1)){
  pred_knn_loocv = knn.cv(train_tot_knn, mydf$Species, k = i, prob = TRUE)

  # Verifica accuratezza del modello
  loocv.err[i] = mean(pred_knn_loocv == mydf$Species)
  }

plot(loocv.err)
which.max(loocv.err)
points(which.max(loocv.err), loocv.err[which.max(loocv.err)], col="red",cex=2,pch=20)

loocv.err[which.max(loocv.err)]

pred_knn_loocv = knn.cv(train_tot_knn, mydf$Species, k = 11, prob = TRUE)

table(pred_knn_loocv, mydf$Species)



###    3    #####
# trovare il miglior K della KNN nella cross validation
```