CLASSIFICAZIONE del TUMORE DUTTALE PANCREATICO tramite BIOMARCATORI

UNIVERSITÀ degli STUDI di MILANO BICOCCA

Lombardi Mattia

***Abstract**: questo report intende analizzare, mediante classificazione supervisionata, il dataset aggiornato che era stato adoperato nello studio della Dott.ssa Debernardi per una previa valutazione del tumore pancreatico (Dec 2020, Journal PLOS Medicine) [[1]]. I metodi che verranno presentati sono il classificatore EDDA e un modello regressivo di misture finite gaussiane (GMM). Sono riportati, con classi bilanciate, individui: sani, aventi una malattia epatobiliare benigna o aventi un adenocarcinoma duttale pancreatico.*

Il dataset è stato scaricato dalla piattaforma Kaggle:
https://www.kaggle.com/datasets/johnjdavisiv/urinary-biomarkers-for-pancreatic-cancer?select=Debernardi+et+al+2020+data.csv

1.INTRODUZIONE

L'adenocarcinoma duttale pancreatico (PDAC) è uno dei tumori più letali. Una volta diagnosticato, il tasso di sopravvivenza a 5 anni è inferiore del 10%. Tuttavia, se diagnosticato per tempo, le probabilità di sopravvivenza e la prognosi possono migliorare. Sfortunatamente molti casi di tumore al pancreas non mostrano sintomi fino a quando non raggiungono stadi più avanzati. Per questo motivo è importante individuare tecniche diagnostiche, non invasive, che possano valutare la presenza del tumore nei primi stadi. Infatti, questo dataset è costituito principalmente da biomarcatori delle urine e uno del plasma.

2.STRUTTURA DATI

Il dataset è composto da 590 osservazioni di 14 variabili. I dati sono stati raccolti da diverse fonti: Barts Pancreas Tissue Bank, University College London, University of Liverpool, Spanish National Cancer Research Center, Cambridge University Hospital, and University of Belgrade.

Le variabili *diagnosis* (31% control, 35% diagnosi benigna e 34% diagnosi tumore duttale pancreatico) e *sex* (49% M e 51% F) sono perfettamente bilanciate, questo suggerisce che sia stato effettuato un *retrospective sampling*. Quest'ultimo prevede di ottenere un dataset come concatenazione di sottocampioni, fin quando non si ottengono classi bilanciate da poter poi adoperare nel *training set*. Pertanto, il dataset va inteso per fini diagnostici e non descrittivi del fenomeno in questione.

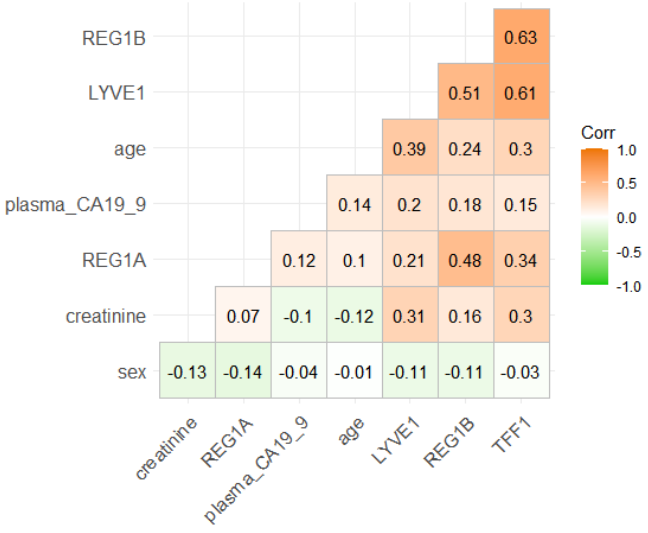
Una presentazione sintetica delle variabili coinvolte nel dataset:

variabili	descrizione
sample_id	stringa che identifica il paziente
patient_cohort	se l'osservazione proviene dal nuovo campione
sample_origin	da quale clinica viene il campione
age	età del paziente
sex	sesso del paziente (1-M;2-F)
diagnosis	(1-control,2-benigno,3-cancro)
stage	(IA, IB, IIA, IIIB, III, IV)
benign_sample	diagnosi per malattia benigna
plasma_CA19_9	anticorpo monoclonale che risulta spesso elevato nei pazienti con tumore pancreatico
creatinine	biomarcatore della funzione dei reni
LYVE1	endotelio dei vasi linfatici
TFF1	"Trefoil Factor 1", biomarcatore coinvolto nella riparazione tessuti
REG1B	proteina associata alla rigenerazione del pancreas
REG1A	"", aggiunta successivamente per valutazione e confronto con REG1B

Purtroppo, le variabili *plasma_CA19_9* e *REG1A* presentano rispettivamente il 40.68% e 48.14% di dati mancanti, che rappresentano il 6.3% del totale. Dato che supera la canonica soglia del 5% non possono essere trascurati. Inoltre, le osservazioni mancanti risultano più frequenti nei pazienti senza diagnosi maligna, quindi i dati mancanti sono NMAR (probabilmente i pazienti non malati non eseguono di routine tali analisi). Pertanto, si è deciso di procedere sia tramite la non considerazione delle variabili (caso A), che tramite imputazione dei dati mancanti (caso B). In tal senso è stata adottata la PPCA (Probabilistic Principal Component Analisys), metodo che assume la normalità delle variabili e, perciò, ben si presta nell'imputazione dei dati mancanti in ambito medico-scientifico [[2]]. Inoltre, è un metodo robusto anche in caso di >15% *missings values.*

3.ANALISI ESPLORATIVA

L'analisi esplorativa è stata effettuata solo sui dati completi, in quanto si preferiva avere una visione più ampia possibile della relazione tra le variabili. Innanzitutto, visualizziamo la matrice di correlazione:

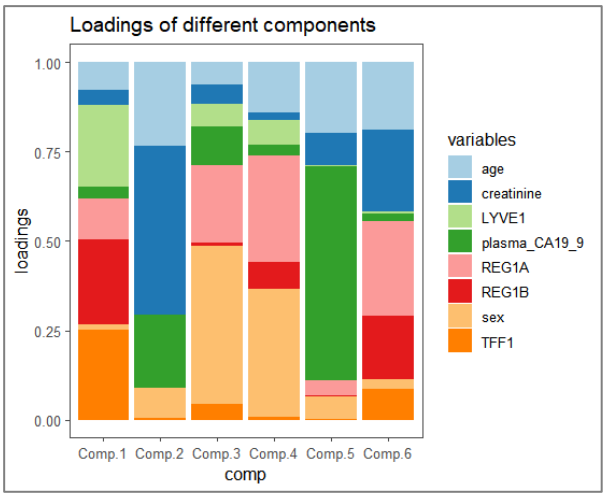


Come possiamo vedere non ci sono variabili particolarmente correlate tra loro ($|\rho| > 0.80$). Pertanto, non vi è necessità di escludere alcuna variabile in quanto ridondante.

PRINCIPAL COMPONENTS ANALYSIS (PCA)

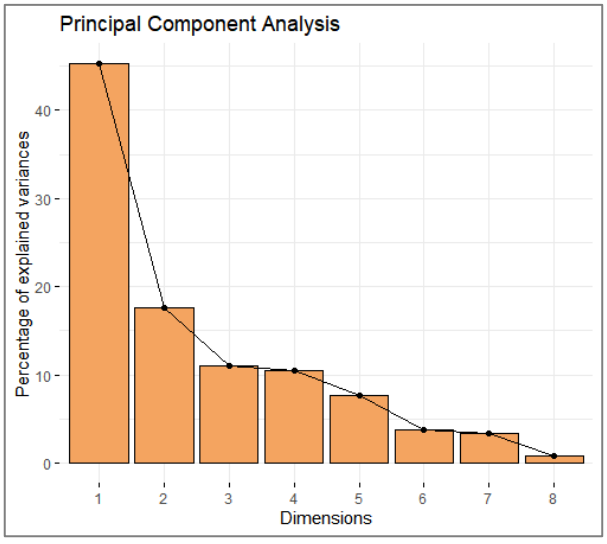
L'analisi delle componenti principali viene solitamente usata per ridurre la dimensionalità di un dataset. In questo caso, invece, si è preferito adoperarla più a titolo esplorativo. Infatti, tale scomposizione permette di comprendere le relazioni e la significatività delle componenti.

Le prime 6 componenti principali spiegano il 91% della variabilità interna. Pertanto, si è deciso di visualizzare i *loadings* di quest'ultime:

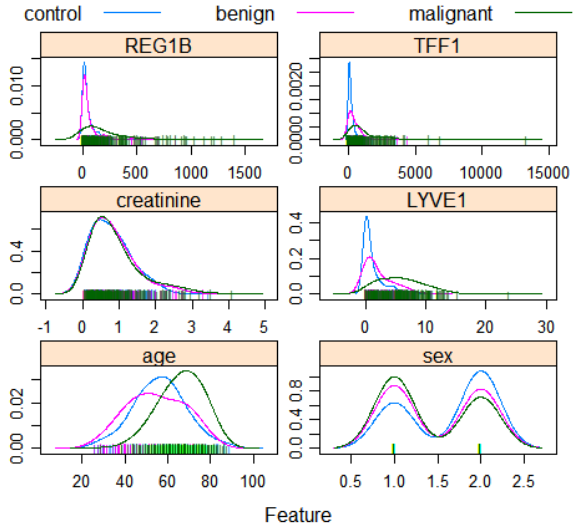


I loadings sono i coefficienti delle combinazioni lineari che definiscono le componenti principali. Quindi restituiscono una misura del contributo di ogni variabile alla componente considerata (1:6).

Si riporta, sottostante, anche il grafico che visualizza la percentuale di varianza spiegata per ogni componente:



Risulta importante anche valutare la distribuzione delle variabili. Come si può notare dal grafico sottostante i pazienti con diagnosi maligna hanno mediamente un'età più avanzata e presentano valori più elevati dei biomarcatori coinvolti nella rigenerazione del pancreas.

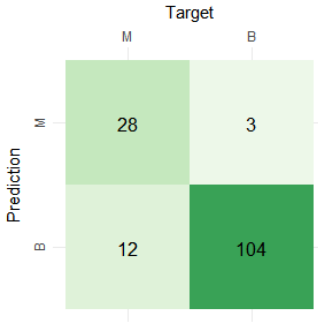


4. EDDA CLASSIFIER

Il classificatore EDDA (Eigenvalue Decomposition Discriminant Analisys) è basato sulla scomposizione della matrice di varianza. Tale metodo di scomposizione viene detto VSO (Volume-Shape-Orientation) e permette la costruzione di una più flessibile struttura di variabilità:

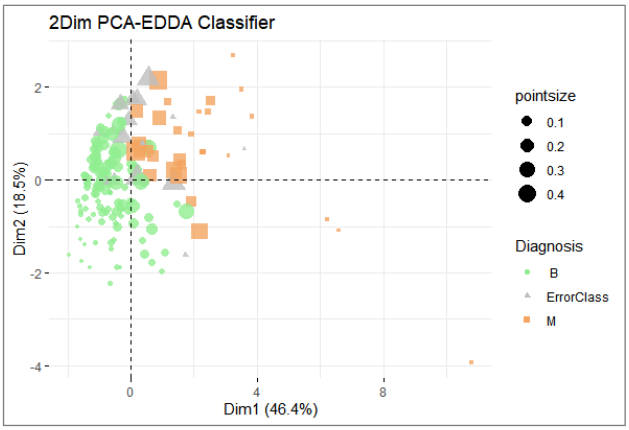
$$\Sigma j = \lambda j D j A j D'$$

Le analisi presentate riguarderanno il dataset che non presenta osservazioni imputate (caso A), in quanto l'aggiunta della covariate aggiuntive non aveva significativamente migliorato il modello. Inoltre, al fine di migliorare la *performance*, si è deciso di suddividere la popolazione in pazienti che non presentano metastasi (B) e pazienti aventi tumore maligno (M). Si procede suddividendo il dataset nella suddivisione training- test (75%-25%). Sono state effettuate 500 iterazioni per trovare il modello che minimizzasse il MER, tramite convalida incrociata (CV-10), e il BIC. Si sono poi ordinati i risultati mediante l'algoritmo *radix sort*. In tal senso il modello migliore è risultato *Gaussian_pk_L_Ck* (**EVV**) , ovvero un modello che presenta struttura di variabilità con volume **e**guale, forma **v**ariabile ed orientamento **v**ariabile. Il secondo miglior modello è *Gaussian_pk_L_C* (EEE), il quale coincide anche con la LDA (Linear Discriminant Analysis), un modello che presenta vincoli sulla matrice di varianza molto stringenti. Si è poi valutata la capacità previsiva del modello sul test-set, confrontando la vera diagnosi con quella stimata dal classificatore. Si è ottenuta la seguente matrice di confusione:



ACCURACY = 89.8%
SENSITIVITY = 89.66%
SPECIFICITY = 90.32%

Nonostante si abbia ottenuto una buona *accuracy* (89.8%), il classificatore non è riuscito a discriminare correttamente tutti i pazienti con metastasi. Purtroppo alcuni valori come la *TFF1* possono risultare molto elevati anche in caso di diagnosi secondarie (principalmente pancreatiti). Infatti si è ottenuto un ARI (Adjusted Rand Index) pari solo al 61%, il che indica che non era presenta una netta differenza tra le due partizioni. Inoltre, l'informazione tra chi aveva una diagnosi secondaria benigna e chi era un soggetto sano è stata perduta. Nel grafico sottostante si rappresentano le osservazioni misclassificate considerando le prime due componenti principali. La grandezza dei punti rappresenta l'incertezza, infatti come possiamo notare le principali classificazioni errate erano contese tra le due nuvole di punti:



5. FINITE MIXTURES of REGRESSION MODEL (GMM)

Date le poche variabili a disposizione ed il fenomeno di studio, solo parzialmente noto, si è deciso di adottare una regressione di modelli mistura. Infatti, questi modelli riescono bene a catturare l'eterogeneità di una o più variabili non osservate (latenti). Legge distribuzione mistura:

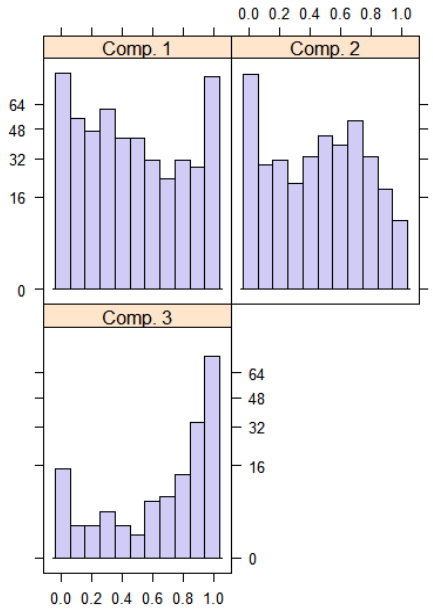
$$f(y \mid \varphi, x) = \sum_{k=1}^{K} \pi_k(x).\, f_k(y \mid \Theta_k, x)$$

In tal caso si considera la risposta y (diagnosi) distribuita normalmente e suddivisa nei k=3 gruppi iniziali (C-B-M). Le covariate corrispondono alle variabili adoperate nella PCA con aggiunta delle osservazioni mancanti imputate (caso B), poiché in tal caso si è ottenuta una miglior classificazione.

diagnosis~sex+age+creatinine+TFF1+LYVE1+ plasma_CA19_9+REG1A+REG1B

È stata utilizzata la libreria *flexmix* di Rstudio ed è stato implementato un Full MeM (Expert Network +Gating Network), ovvero sono stati modellizzati sia i parametri φ_k sia i pesi π_k. Inoltre il modello è stato scelto tramite 1000 iterazioni con il criterio di minimizzare sia EIC che BIC.

Rootogram of posterior probabilities > 1e-04



ICL	BIC	EIC
1184	925	0.62

Le iterazioni servono anche come meccanismo di controllo, poiché l'algoritmo E-M tende ad incorrere in massimi locali. Come possiamo vedere dal grafico delle probabilità a posteriori, le quali rappresentano il grado di incertezza con il quale sono state allocate ai gruppi le etichette, e dalla distanza Kullback-Leiber, le componenti B-M sono quelle tra loro più vicine e quindi più difficili da collocare.

Distanza K-L	
Comp(1-2 ; B-M)	2455.1
Comp(2-3 ; M-C)	150142
Comp(1-3 ; B-C)	75448.6

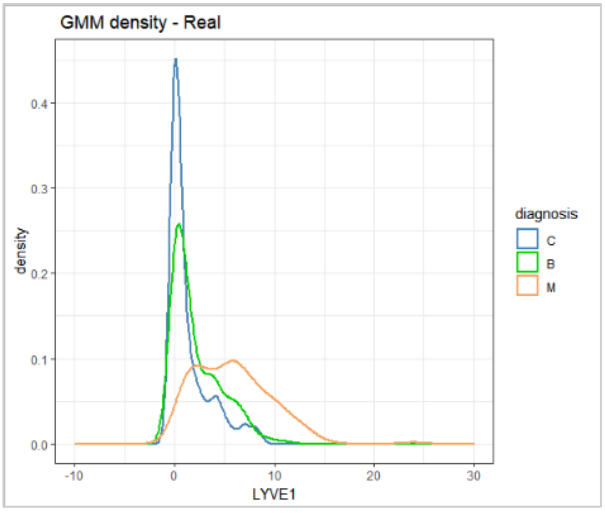
Gli *outliers* delle variabili *plasma_CA19_9* e *TFF1* sono stati rimossi in modo tale che il modello potesse differenziare maggiormente le componenti (rimangono 534 osservazioni). Qui di seguito visualizziamo la matrice di confusione ottenuta:

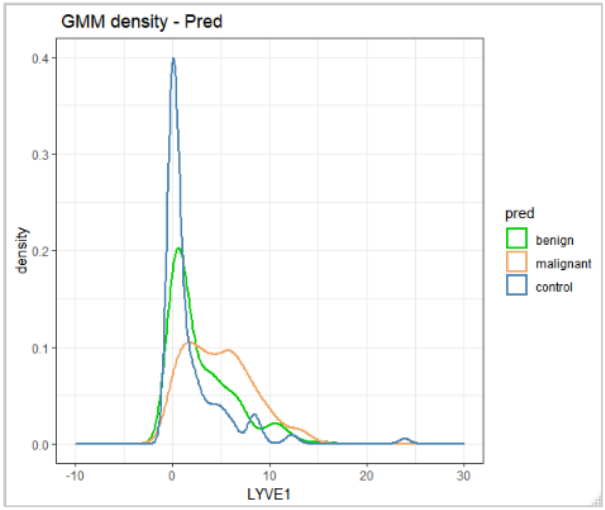
ACCURACY = 85.58%
SENSITIVITY(M) = 89.25%
SPECIFICITY(M) = 90.52%

Gli errori di primo e secondo tipo sono stati riportati solamente per la diagnosi maligna (M), in quanto è la componente che principalmente ci interessa discriminare. Notare che nessun paziente del gruppo di controllo è stato classificato come malato, infatti come possiamo notare la distanza di KL è molto elevata tra M-C.

Dall'analisi della regressione otteniamo che: la componente C è stata modellizzata da *sex(***),age(***),LYVE1(***)* e *REG1B(*)*. Mentre la componente M :*intercept(***), age(**), TFF1(**), sex, plasma_CA19_9, LYVE1, REG1A*. La componente B, invece, solamente da: *intercept(***) e LYVE1.*Gli asterischi corrispondono al livello di significatività dei coefficienti di regressione dei rispettivi modelli. (***- 0 ; **- 0.01 ; * - 0.05)

I modelli di misture gaussiane possono essere adoperati quando risulta difficile catalogare le osservazioni come appartenenti ad uno o all'altro gruppo, poiché riescono a classificare le unità in base al modello mistura, in tal caso normale, che meglio descrive ogni componente. Per valutare graficamente la bontà della regressione di misture gaussiane possiamo visualizzare il grafico della densità della variabile *LIVE1*per le etichette previste e poi confrontarla con la distribuzione di frequenza di quelle reali (lo scambio di etichette è dovuto all'algoritmo E-M). Grafici sottostante:





6.CONCLUSIONI

I metodi *model-based* presentati sono risultati più che adeguati a prevedere e modellizzare, con una buona accurratezza e precisione, la diagnosi di PDAC (adenocarcinoma duttale pancreatico).

Inoltre, in tal caso non avevamo a disposizione un grande numero di osservazioni (590), quindi tecniche non parametriche avrebbero potuto dare performance inferiori, oltre a non poterci restituire un'interpretazione della significatività delle variabili.

In realtà anche il classificatore EDDA avrebbe giovato di un dataset più grande e completo,in quanto avrebbe avuto un più ampio training set. Pertanto risulterebbe utile continuare a collezionare ulteriori dati riguardanti la patologia, perché ripetere le analisi su un dataset più fornito potrebbe migliorare di molto le performance.

Purtroppo raggiungere un'*accuracy* pari al 100% risulterebbe difficile in ogni caso, poiché alcuni malati di PDAC non presentano analisi dei biomarcatori notevolemente alterate rispetto ai pazienti sani o con diagnosi secondaria.
In tal senso approfondire con altre libraries, o metodologie di convergenza per l'algoritmo E-M, le regressioni di misture finite gaussiane (GMM) potrebbe risultare interessante, in quanto sono un modello che riesce a cogliere la presenza di variabili latenti. Pertanto potrebbero risultare particolarmente utili in questo specifico caso. Inoltre adoperare modelli statistici più avanzati, come quelli Bayesiani, potrebbe risultare interessante e dare notevoli spunti ai fini della classificazione.

REFERENCES

[[1]]
https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1003489

[[2]]

https://www.sciencedirect.com/science/article/pii/S2352914819302783

In dedica e memoria di mio Nonno Luciano

CODICE ADOPERATO per GRAFICI ed ANALISI

(software adoperato: Rstudio)

mat: 837193

```r
#Mattia Lombardi#

#project of Computational Statistics:
#Model based Classification of pancreatic cancer

#libraries:
library(tidyverse)
library(ggplot2)
library(mclust)
library(visdat) #missed data visualization
library(ggcorrplot)
library(GGally)
library(ggthemes)
library(factoextra) #pca library

#----------------------------------------
#PCA and Exploration Data Analysis (EDA)
#----------------------------------------

setwd("C:/Users/39349/OneDrive/Desktop/UNIMIB/dataset")

data<-read.csv("pancreatic_cancer.csv")
#visualize data
str(data)
View(data)

#cambio in numerica la variabile sex M-F
data$sex<-ifelse(data$sex=="M",1,2)
table(data$diagnosis)/590
table(data$sex)/590
#proportion are ok

#missing data
anyNA(data) #TRUE
vis_miss(data)

#purtroppo le variabili REG1A e plasma_CA19_9
#hanno molti valori mancanti
#per fare un analisi preliminare delle PCA
#consideriamo solamente le 209 osservazioni complete

data_comp<-data[complete.cases(data),]
#considero variabili solo numeriche
data_num<-data_comp[,-c(1,2,3,6,7,8)]
data_lab<-as.factor(data_comp[,6])
table(data_lab)/209

#possiamo notare che i missing values provenivano
#prevalentemente dal gruppo di controllo o con diagnosi benigna
#dati mancanti NMAR
str(data_num)

#correlazione
ggcorrplot(cor(data_num), hc.order = TRUE, type = "lower",lab = TRUE,
           colors = c("green3", "white", "darkorange2"))
```

```r
#PCA
pca<-princomp(data_num,cor=T)
pca

#rappr. della % di varianza spiegata dalle components
fviz_eig(pca, barfill = "sandybrown",barcolor="black",main = "Principal Component Analysis")
sum(pca$sdev[1:6]^2)/8
#le prime 6 components spiegano 91% varianza within

#grafici non presenti nel report:
#1)rappr. del grafico rappresentante le 2 componenti principali suddivise per sesso
groups <- as.factor(ifelse(data_num$sex==1,"M","F"))

fviz_pca_ind(pca,
             col.ind = groups,
             palette = c("#FC4E07","#00AFBB" ),
             addEllipses = TRUE,
             legend.title = "Sex",
             repel = TRUE,
             title=c("2Dim PCA-Sex")
)

#2)rappresentazione grafica 2PCA dimensions - creatinine
fviz_pca_ind(pca, col.ind=data_num$creatinine,
             legend.title = "Creatinine",
             repel = TRUE,
             title=c("2Dim PCA-Creatinine")
)+scale_color_gradient2(low="white", mid="slateblue",
                        high="orange", midpoint=0.6)

#costruisco il grafico dei loadings
scores<-pca$loadings[,1:6]^2
scores
data_scores<-as.vector(scores)

comp<- c ( rep("Comp.1",8),rep("Comp.2",8),rep("Comp.3",8),rep("Comp.4",8),rep("Comp.5",8),rep("Comp.6",8))
variables<-c("age","sex","plasma_CA19_9","creatinine","LYVE1","REG1B","TFF1","REG1A")
data_loadings<-data.frame(comp,variables,data_scores)
data_loadings

#grafico loadings
ggplot(data_loadings, aes(fill=variables, y=data_scores, x=comp)) +
  geom_bar(position="stack", stat="identity")+
  labs(title = "Loadings of different components",y="loadings")+
  scale_fill_brewer(palette = "Paired")+
  theme_few()
data_num
#mostro le funz di densità delle variabili
#suddivise per ogni componente
library(caret)
levels(data_lab)<-c("control","benign","malignant")

featurePlot(data_num[,-c(3,8)],data_lab,plot="density",
            scales=list(x = list(relation="free"),y = list(relation="free")),
            adjust = 1.5,pch="|",auto.key = list(columns = 3))
```

```r
#-----------------
#EDDA CLASSIFIER
#-----------------

#rileggo i dati per evitare eventuali errori
data<-read.csv("pancreatic_cancer.csv")


#data manipulation:
data$sex<-ifelse(data$sex=="M",1,2)

#modifico le etichette in diagnosi senza metastasi e con cancro
data$diagnosis<-ifelse(data$diagnosis %in% c(1,2),1,2)

#considero solo le variabili complete e numeriche
dada<-data[,-c(1,2,3,6,7,8,9,14)]
#etichette reali
data_lab<-as.factor(data$diagnosis)

#per ottenere sempre stessi risultati
set.seed(123)

#training e test set
test_lab = sample(1: 590 ,trunc(0.25* 590),replace = F)

train<-as.data.frame(dada[-test_lab,])
data_train_lab<-data_lab[-test_lab]

#dataframe per inserire risultati
results = as.data.frame ( matrix ( nrow = 300 , ncol = 2))
colnames(results)=c("model","cv")

library(Rmixmod)
clas.train=mixmodLearn(train,data_train_lab,
                       models = mixmodGaussianModel(family="all",equal.proportions=FALSE),
                       criterion="BIC")
clas.train

for ( i in 1:300){
  clas.train=mixmodLearn(train,data_train_lab,
                         models = mixmodGaussianModel(family="all",equal.proportions=FALSE),
                         criterion=c("CV","BIC"))

  model=clas.train@bestResult@model
  cv=clas.train@bestResult@criterionValue[1]
  results[i,]=c(model,cv)
}

#ordino i risultati
results$cv = as.numeric ( results$cv )
results<-results[order(results$cv,method="radix"),]

#seleziono miglior modello
best.mod = mixmodLearn (train,data_train_lab,
                        models = mixmodGaussianModel(listModels=results$model[1]))
best.mod["bestResult"]
```

```
169
170 #prediction
171 pred.values=mixmodPredict(data=dada[test_lab,],classificationRule=best.mod["bestResult"])
172
173 Uncertainty=apply(pred.values@proba,1,min)
174 types=factor(pred.values@partition,levels =c(1,2),labels = c("B","M"))
175
176 act = factor ( data_lab [ test_lab ] , levels = c ("1","2") ,
177                labels = c (" B ","M"))
178
179 prd = factor ( pred.values@partition , levels = c ("1","2") ,
180                labels = c (" B ","M"))
181
182
183 library(caret)
184 conf.matrix = confusionMatrix ( data = act , reference = prd )
185 conf.matrix
186
187 tab_conf<-as_tibble(conf.matrix$table)
188
189 #library per costruire una confuzion matrix carina
190 library(cvms)
191
192 matrix<-table(tibble("target"=act,
193                      "prediction"=prd))
194
195 plot_confusion_matrix(as_tibble(matrix),
196                       target_col = "target",
197                       prediction_col = "prediction",
198                       counts_col = "n",
199                       add_normalized = FALSE,
200                       add_col_percentages = FALSE,
201                       add_row_percentages = FALSE,
202                       palette = "Greens")
203
204 #(ADI)
205 adjustedRandIndex (act,prd)
206
207 #visualizzo grafico con unità misclassificate e livello incertezza
208 library(mclust)
209 mis<-classError(act,prd)
210
211 #creo un dataset contenente variabili misclassified
212 error_data<-data[test_lab[mis$misclassified],]
213 error_data<-cbind(error_data,prd[mis$misclassified])
214 #ppca graphic of predicted variables
215 data_pred<-dada[test_lab,]
216
217 pca<-princomp(data_pred,cor=T)
218 prd = factor ( pred.values@partition , levels = c ("1","2") ,
219                labels = c (" B ","M"))
220
221 miss_class<-mis$misclassified
222 data_pred$diagnosis<-as.character(prd)
223 data_pred$diagnosis[miss_class]<-"ErrorClass"
224 data_pred$diagnosis<-as.factor(data_pred$diagnosis)
```

```r
#grafico tramite factoextra che visualizza le prime due PCA
fviz_pca_ind(pca, col.ind=data_pred$diagnosis,
             geom = c("point"),alpha=0.8,
             pointsize=Uncertainty,
             legend.title = "Diagnosis",
             repel = TRUE,
             title=c("2Dim PCA-EDDA Classifier"))+
  scale_color_manual(values = c("palegreen2","gray","sandybrown"))

#-----------------------------------------------
#FINITE MIXTURE of REGRESSION MODEL (GAUSSIAN)
#-----------------------------------------------

#rileggo i dati
data<-read.csv("pancreatic_cancer.csv")

#data manipulation
data$sex<-ifelse(data$sex=="M",1,2)
#in questo caso consideriamo k =3

#PPCA imputation method
library(missMDA)
#ppca funz solo con num variabili
#ipotizza normalità delle variabili

data_num<-data[-c(1,2,3,6,7,8)]
nb<-estim_ncpPCA(data_num,scale=T)
comp<-imputePCA(data_num,ncp = 5,scale=T)

data.pca<-abs(as.data.frame(comp$completeObs))
#View(data.pca)

data_num2<-data[,-c(1,2,3,7,8,9,14)]

#data visualization (non presente nel report)
#la variabilità aumenta man mano che la diagnosi peggiora
ggplot(data=data_num2, mapping=aes(x=creatinine,y=LYVE1))+
  geom_point(aes(colour=diagnosis))+
  geom_smooth(method="lm", se=F, size=1.5)+
  theme_light()

library(mixtools)
library(flexmix)
# flexmix(formula, data, k = NULL, cluster = NULL,
#         model = NULL, concomitant = NULL,.....)

#aggiungo la variabile diagnosis
#nostra variabile risposta della regressione
datapi<-cbind(data[c(6)],data.pca)

#rimuovo outliers
data_ben=datapi[datapi$diagnosis==c(2),]

(outliers_tf<-boxplot.stats(data_ben$TFF1)$out)
(outliers_pla<-boxplot.stats(data_ben$plasma_CA19_9)$out)
```

```r
data_fin<-datapi[-which(data_ben$TFF1 %in% outliers_tf),]
data_fin2<-data_fin[-which(data_ben$plasma_CA19_9 %in% outliers_pla),]
data_fin3<-data_fin2[-which(data_ben$plasma_CA19_9 %in% outliers_pla),]

datapi<-data_fin3
#str(datapi)
#algoritmo controllo E-M
itermax <- 1000

#entropia
eicval=1
eics<-matrix(nrow=itermax,ncol=2)
#bic
bicval <- Inf
bics<-matrix(nrow=itermax,ncol=2)
#n.b: da considerare abs(), potrebbe assumere valori negativi

#meccanismo di controllo per EM
#(tempistiche: circa 5 min)

#per ridurre il tempo ho confrontato solamente entropia
#tramite un metodo di MC, non molto ufficiale,
#se EIC si discosta molto dalla soglia 0.6 la classificazione
#inizia ad accorpare le due componenti M-B

for (iter in 1: itermax)
{
  fit <-flexmix(diagnosis~.,data=datapi,k=3)
  eics[iter,]<-c(iter,EIC(fit))
  if (EIC(fit)<eicval)
  {
    eicval<-EIC(fit)
    bestfit <- fit
  }
}

#results:
summary(bestfit)
plot(bestfit)
BIC(bestfit)
EIC(bestfit)
ICL(bestfit)  #ICL=entropia+BIC
KLdiv(bestfit) #distance Kullback Leiber

fit=bestfit

lab<-as.factor(datapi[,1])
pred<-as.factor(fit@cluster)

#riaggiusto i livelli etichette
levels(pred)<-c(2,3,1)
#pred
#lab
levels(pred)<-c("B","M","C")
levels(lab)<-c("C","B","M")
```

```r
conf.matrix = confusionMatrix ( data = lab, reference = pred)
conf.matrix
#viasualizza graficamente confuzion matrix
tab_conf<-as_tibble(conf.matrix$table)

matrix<-table(tibble("target"=lab,
                     "prediction"=pred))

plot_confusion_matrix(as_tibble(matrix),
                      target_col = "target",
                      prediction_col = "prediction",
                      counts_col = "n",
                      add_normalized = FALSE,
                      add_col_percentages = FALSE,
                      add_row_percentages = FALSE,
                      palette = "Oranges")

adjustedRandIndex(lab,pred)
fit@components
#varianza delle components
#come si notava dal grafico iniziale
#sono più elevate man mano che la diagnosi è più grave
(sigma<-parameters(fit)[10,])

#analisi della regressione
summary(refit(fit))

fit@components
fit@df #gradi libertà
#parametri
par<-parameters(besttfit)

#creo dataset con dati predicted
data_pred<-cbind(datapi[,-1],pred)
levels(pred)<-c("B","M","C")

#visualizzo funzione di densità per etichette previste
#considero LIVE1 in quantouna delle variabili più significative
ggplot(data_pred,aes(x=LYVE1,color=pred))+
  geom_density(lwd=1)+
  xlim(-10,30)+
  scale_color_manual(values=c("green3", "sandybrown", "steelblue"))+
  labs ( title = " FMM density - Pred")+
  theme_bw()


datapi$diagnosis<-as.factor(datapi$diagnosis)
levels(datapi$diagnosis)<-c("C","B","M")
#visualizzo distribuzione dei gruppi reali
ggplot(datapi,aes(x=LYVE1,color=diagnosis))+
  geom_density(lwd=1)+
  xlim(-10,30)+
  theme_bw()+
  scale_color_manual(values = c("steelblue","palegreen2","sandybrown"))+
  labs ( title = " FMM density - Real")
```