UNIVERSITA' DEGLI STUDI DI CATANIA

DIPARTIMENTO DI ECONOMIA E IMPRESA

CORSO DI LAUREA MAGISTRALE IN FINANZA AZIENDALE

Giuseppe Antonio Sanfilippo

CLUSTER ANALYSIS APPLICATA ALLE VARIABILI DI UN DATASET MUSICALE

MODELLI STATISTICI PER L'ECONOMIA E LA FINANZA

Prof. Di Mari Roberto

ANNO ACCADEMICO 2019 - 2020

INTRODUZIONE AL PROGETTO

L'oggetto di studio del seguente elaborato è l'analisi statistica multivariata applicata ad un dataset contente un gran numero di osservazioni. In particolare, utilizzando le funzionalità del software RStudio, ho svolto una cluster analysis (sfruttando anche l'Analisi delle Componenti Principali) cercando di ricondurre elementi tra loro eterogenei in più sottoinsiemi tendenzialmente omogenei e mutuamente esaustivi, applicando rappresentazioni grafiche in grado di mostrare in maniera semplice la classificazione eseguita.

PRESENTAZIONE DEL DATASET

Il dataset preso in considerazione[1] contiene un gran numero di osservazioni concernenti l'ambito musicale: nel dettaglio sono presenti 232725 tracce musicali a cui la piattaforma di streaming audio Spotify ha applicato, tramite specifiche procedure di *Application Programming Interfaces,* 18 variabili sia quantitative che qualitative finalizzate ad approfondire le caratteristiche fondamentali di ciascuna canzone.

Una volta aperto su RStudio il file "music.csv" tramite la funzione:

```
music <- read.csv("music.csv",header = TRUE)
```

Tramite il comando `str(music)` è possibile dare un'occhiata preliminare a quelle che sono le variabili che compongono il dataset.

```
'data.frame':    232725 obs. of  18 variables:
 $ ï..genre        : Factor w/ 27 levels "A Capella","Alternative",..: 16 16 16 16 16 16
16 16 16 16 ...
 $ artist_name     : Factor w/ 14564 levels "'Til Tuesday",..: 5283 8366 6575 5283 4140
5283 8366 7434 2465 7469 ...
 $ track_name      : Factor w/ 148615 levels "' Cello Song",..: 20191 96046 34319 33138
93914 71569 99298 72873 52992 72167 ...
 $ track_id        : Factor w/ 176774 levels "00021Wy6AyMbLP2tqij86e",..: 4971 4768 5608
8233 10160 12589 13641 14649 17254 19465 ...
 $ popularity      : int  0 1 3 0 4 0 2 15 0 10 ...
 $ acousticness    : num  0.611 0.246 0.952 0.703 0.95 0.749 0.344 0.939 0.00104 0.319 .
..
 $ danceability    : num  0.389 0.59 0.663 0.24 0.331 0.578 0.703 0.416 0.734 0.598 ...
 $ duration_ms     : int  99373 137373 170267 152427 82625 160627 212293 240067 226200 1
52694 ...
 $ energy          : num  0.91 0.737 0.131 0.326 0.225 0.0948 0.27 0.269 0.481 0.705 ...
 $ instrumentalness: num  0 0 0 0 0.123 0 0 0 0.00086 0.00125 ...
 $ key             : Factor w/ 12 levels "A","A#","B","C",..: 5 10 4 5 9 5 5 10 4 11 ...
 $ liveness        : num  0.346 0.151 0.103 0.0985 0.202 0.107 0.105 0.113 0.0765 0.349
...
 $ loudness        : num  -1.83 -5.56 -13.88 -12.18 -21.15 ...
 $ mode            : Factor w/ 2 levels "Major","Minor": 1 2 2 1 1 1 1 1 1 1 ...
 $ speechiness     : num  0.0525 0.0868 0.0362 0.0395 0.0456 0.143 0.953 0.0286 0.046 0.
0281 ...
 $ tempo           : num  167 174 99.5 171.8 140.6 ...
 $ time_signature  : Factor w/ 5 levels "0/4","1/4","3/4",..: 4 4 5 4 4 4 4 4 4 4 ...
 $ valence         : num  0.814 0.816 0.368 0.227 0.39 0.358 0.533 0.274 0.765 0.718 ...
```

[1] Fonte del dataset: https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db

Sul sito di spotify.com è possibile reperire il significato attribuito a ciascuna delle variabili prese in considerazione[2].
Nelle operazioni di cluster e PCA che voglio portare a termine, le variabili che maggiormente mi interessano ai fini dell'analisi sono:

Acustica: Misurata in un intervallo di confidenza che va da 0,0 (basso grado d'acustica) ad 1,0 (alta confidenza ed ottima acustica).

Ballabilità: Descrive quanto una traccia sia adatta per ballare, prendendo in considerazione una serie di elementi musicali combinati tra loro (tra cui tempo, stabilità del ritmo, forza del battito e regolarità generale). Ad un valore di 0,0 corrisponde una scarsa ballabilità, mentre ad 1,0 corrisponde il massimo adattamento.

Energia: Rappresenta una misura percettiva di intensità e attività. In genere, brani appartenenti ai generi rock e metal hanno valori più vicini all'1,0 mentre canzoni classiche e musica d'ambiente hanno valori più bassi nella scala di misurazione.

Vividezza: Questa variabile rileva la presenza di pubblico nella registrazione. Valori di vividezza più elevati rappresentano una maggiore probabilità che la traccia sia stata eseguita dal vivo. Un valore superiore a 0,8 fornisce una forte probabilità che la traccia sia stata registrata durante una performance live.

Rumorosità: Tiene conto del volume generale di una traccia misurato in decibel. I valori di rumorosità sono mediati su tutta la durata del brano e sono utili per confrontare il volume relativo delle tracce. I valori tipici sono compresi tra -60 e 0 decibel.

Valenza: Parametro che descrive la positività musicale trasmessa da una traccia. I brani con alta valenza suonano più positivi (felici, allegri, euforici), mentre quelli con bassa valenza suonano più negativi (tristi, scoraggianti, arrabbiati).

Per poter effettuare un'analisi più efficiente, tuttavia, terrò in considerazione anche le variabili relative al Genere, all'Artista, al Nome della Canzone e alla **Popolarità**.

Dato inoltre che la variabile Popolarità è espressa in come *Integer*, per poter procedere nell'operazione mi interessa convertirla in valori numerici (tramite la funzione `as.numeric`), per evitare di riscontrare errori nelle operazioni successive.

[2] https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/

Ho quindi caricato il pacchetto "`dplyr`" per poter manipolare più agevolmente il dataframe ed ho effettuato la conversione attraverso la funzione "`mutate`".

```
library(dplyr)
music <- music %>%
mutate(popularity = as.numeric(popularity))
```

Adesso devo selezionare le variabili che voglio considerare ai fini dell'analisi cluster e PCA. Prima di fare ciò, ho preferito prima rinominarle per facilitare le operazioni successive, usando il comando `names`:

```
names(music)[1]<-paste("genere")
names(music)[2]<-paste("artista")
names(music)[3]<-paste("nome_traccia")
names(music)[5]<-paste("popolarità")
names(music)[6]<-paste("acustica")
names(music)[7]<-paste("ballabilità")
names(music)[9]<-paste("energia")
names(music)[12]<-paste("vividezza")
names(music)[13]<-paste("rumorosità")
names(music)[18]<-paste("valenza")
```

Adesso, tramite il comando `select` ho scelto singolarmente le variabili utili, trasferendole in un nuovo dataset che ho chiamato music2.

```
music2 <- music %>%
select(genere,artista,nome_traccia,popolarità,acustica,ballabilità,energia,vivid
ezza,rumorosità,valenza)
str(music2)
```

```
'data.frame':    232725 obs. of  10 variables:
 $ genere      : Factor w/ 27 levels "A Capella","Alternative",..: 16 16 16 16 16 16 16
16 16 16 ...
 $ artista     : Factor w/ 14564 levels "'Til Tuesday",..: 5283 8366 6575 5283 4140 5283
8366 7434 2465 7469 ...
 $ nome_traccia: Factor w/ 148615 levels "' Cello Song",..: 20191 96046 34319 33138 9391
4 71569 99298 72873 52992 72167 ...
 $ popolarità  : num  0 1 3 0 4 0 2 15 0 10 ...
 $ acustica    : num  0.611 0.246 0.952 0.703 0.95 0.749 0.344 0.939 0.00104 0.319 ...
 $ ballabilità : num  0.389 0.59 0.663 0.24 0.331 0.578 0.703 0.416 0.734 0.598 ...
 $ energia     : num  0.91 0.737 0.131 0.326 0.225 0.0948 0.27 0.269 0.481 0.705 ...
 $ vividezza   : num  0.346 0.151 0.103 0.0985 0.202 0.107 0.105 0.113 0.0765 0.349 ...
 $ rumorosità  : num  -1.83 -5.56 -13.88 -12.18 -21.15 ...
 $ valenza     : num  0.814 0.816 0.368 0.227 0.39 0.358 0.533 0.274 0.765 0.718 ...
```

Il nuovo dataset presenta lo stesso numero di osservazioni, ma solo 10 variabili.

```
summary(music2)
```

```
 genere              artista                          nome_traccia
 Comedy    :  9681   Giuseppe Verdi          :  1394   Home    :   100
 Soundtrack:  9646   Giacomo Puccini         :  1137   You     :    71
 Indie     :  9543   Kimbo Children's Music  :   971   Intro   :    69
 Jazz      :  9441   Nobuo Uematsu           :   825   Stay    :    63
 Pop       :  9386   Richard Wagner          :   804   Wake Up:     59
 Electronic:  9377   Wolfgang Amadeus Mozart:   800   Closer  :    58
 (Other)   :175651   (Other)                 :226794   (Other):232305

 Acustica          ballabilità       energia             vividezza
 Min.   :0.0000   Min.   :0.0569   Min.   :2.03e-05   Min.   :0.00967
 1st Qu.:0.0376   1st Qu.:0.4350   1st Qu.:3.85e-01   1st Qu.:0.09740
 Median :0.2320   Median :0.5710   Median :6.05e-01   Median :0.12800
 Mean   :0.3686   Mean   :0.5544   Mean   :5.71e-01   Mean   :0.21501
 3rd Qu.:0.7220   3rd Qu.:0.6920   3rd Qu.:7.87e-01   3rd Qu.:0.26400
 Max.   :0.9960   Max.   :0.9890   Max.   :9.99e-01   Max.   :1.00000

 rumorosità         valenza          popolarità
 Min.   :-52.457   Min.   :0.0000   Min.   :0.00
 1st Qu.:-11.771   1st Qu.:0.2370   1st Qu.:29.00
 Median : -7.762   Median :0.4440   Median :43.00
 Mean   : -9.570   Mean   :0.4549   Mean   :41.13
 3rd Qu.: -5.501   3rd Qu.:0.6600   3rd Qu.:55.00
 Max.   :  3.744   Max.   :1.0000   Max.   :100.00
```

ANALISI CLUSTER E PCA

Prima di poter procedere con l'analisi cluster e PCA, è necessario standardizzare le variabili che voglio tenere in considerazione, tenendo conto anche della popolarità.

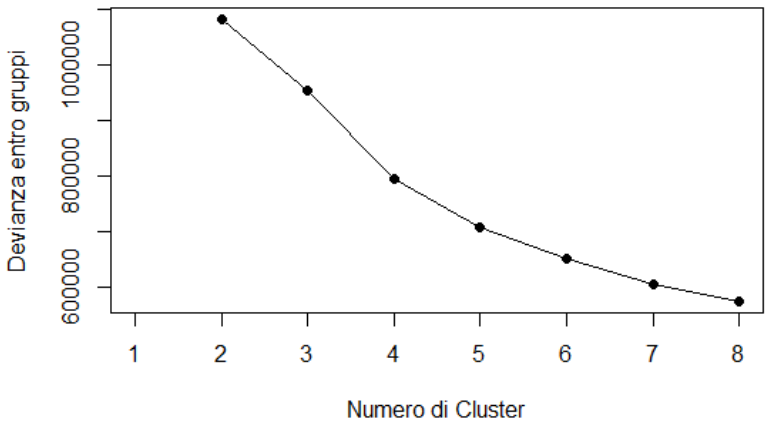
```
standard <- as.data.frame(scale(music2[,c(4:10)]))
```

```
summary(standard)
```

```
   popolarità         acustica         ballabilità          energia
 Min.   :-2.2610   Min.   :-1.0389   Min.   :-2.68019   Min.   :-2.1671
 1st Qu.:-0.6667   1st Qu.:-0.9329   1st Qu.:-0.64310   1st Qu.:-0.7058
 Median : 0.1029   Median :-0.3849   Median : 0.08963   Median : 0.1292
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
 3rd Qu.: 0.7626   3rd Qu.: 0.9963   3rd Qu.: 0.74154   3rd Qu.: 0.8200
 Max.   : 3.2365   Max.   : 1.7686   Max.   : 2.34168   Max.   : 1.6247
   vividezza          rumorosità         valenza
 Min.   :-1.0356   Min.   :-7.1500   Min.   :-1.74924
 1st Qu.:-0.5932   1st Qu.:-0.3670   1st Qu.:-0.83793
 Median :-0.4388   Median : 0.3014   Median :-0.04198
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
 3rd Qu.: 0.2471   3rd Qu.: 0.6784   3rd Qu.: 0.78858
 Max.   : 3.9591   Max.   : 2.2196   Max.   : 2.09595
```

Utilizzando poi il metodo di Elbow, ho determinato il numero appropriato di cluster nell'insieme di dati preso in considerazione, tramite l'iterazione dell'algoritmo *K-means* per diversi valori di K**,** e calcolando la somma delle distanze al quadrato tra ogni centroide ed i punti del proprio cluster. La logica è quella di definire i cluster in modo tale che la variazione totale all'interno del cluster (o la somma totale all'interno del cluster di quadrati, WSS) sia ridotta al minimo. Il WSS totale misura la compattezza del clustering e l'obiettivo è renderlo il più piccolo possibile.

```
WSS <- function(data, maxCluster = 8) {
SSW <- (nrow(data) - 1) * sum(apply(data, 2, var))
SSW <- vector()
for (i in 2:maxCluster) {SSW[i] <- sum(kmeans(data, centers = i)$withinss)}
plot(1:maxCluster, SSW, type = "o", xlab = "Numero di Cluster", ylab = "Devianza
entro gruppi", pch=19)}
WSS(standard)
```



Dal “gomito” si deduce che il numero ideale di cluster da considerare è 4.

Procedo dunque col k-means clustering, generando un semplice set di dati con 4 cluster ben separati. I dati sono ottenuti generando numeri casuali normali tramite il comando `set.seed(n)`, con 20 configurazioni iniziali (`nstart`).

```
set.seed(200)
kmeans1 <- kmeans(standard,4,nstart = 20)
kmeans_cluster <- kmeans1$cluster
kmeans_centroidi <- data.frame(kmeans1$centers,cluster =
rownames(kmeans1$centers))
```

K-means (centroidi)

popolarità	acustica	ballabilità	energia	vividezza	rumorosità	valenza	cluster
0.52722432	-0.5086709	-0.02486497	0.3141571	-0.2314361	0.4361297904	-0.43628431	1
0.01196138	-0.4132294	0.70132073	0.4493017	-0.2316137	0.4119535865	1.04726510	2
-0.63620123	0.4462561	-0.12229762	0.4303142	2.6883682	0.0002755119	-0.04647519	3
-0.68515611	1.3699954	-1.02578561	-1.4423635	-0.3159937	-1.4220445984	-0.88725258	4

Il k-means si basa sui centroidi. Un centroide è un punto appartenente allo spazio delle features che media le distanze tra tutti i dati appartenenti al cluster ad esso associato.

I centroidi, proprio per le loro caratteristiche, sono come dei baricentri del cluster, e di conseguenza non sono punti del dataset.

Analisi delle Componenti Principali (PCA)

Eseguo la PCA con le variabili standardizzate in precedenza.

```
pca <- prcomp(standard[,1:7],center = T)
```

Otterrò in tutto tante componenti principali (PC) quante sono le variabili osservate, ed ognuna sarà ottenuta come combinazione lineare a varianza massima sotto il vincolo di non correlazione con tutte le precedenti.

Tramite la funzione `print(pca)` è possibile osservare le deviazioni standard delle componenti e la matrice di rotazione (con i *variable loadings*).

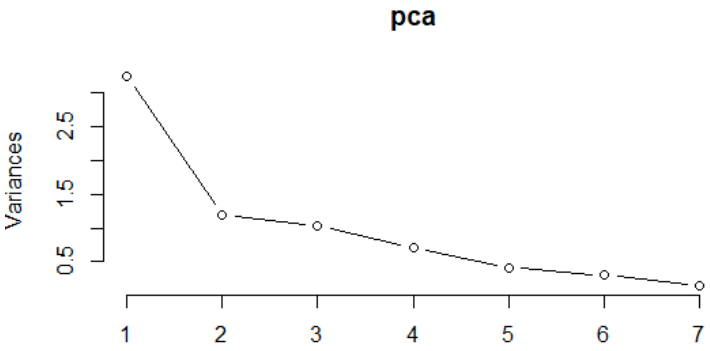
```
Standard deviations (1, .., p=7):
[1] 1.7976468 1.0898585 1.0130055 0.8365787 0.6436672 0.5492793 0.3723103

Rotation (n x k) = (7 x 7):
                PC1         PC2         PC3         PC4          PC5          PC6         PC7
popolarità  0.2617872 -0.52951340  0.2955831 -0.65684523  0.359102802 -0.001116006  0.05754889
acustica   -0.4612422  0.09909505 -0.2510155 -0.26025162  0.171652535 -0.747051850  0.24319768
ballabilità 0.3565010 -0.06148327 -0.5576201 -0.38349550 -0.619501827  0.004159958  0.16528172
energia     0.4798701  0.23361456  0.2515753  0.22849930  0.084533596 -0.165261194  0.75179095
vividezza   0.0127382  0.78637589  0.2382604 -0.52993168  0.007317283  0.158004401 -0.13724494
rumorosità  0.4927410  0.04833845  0.1917852  0.12791077 -0.124972815 -0.615857046 -0.55392137
valenza     0.3440914  0.17526016 -0.6179336  0.06598454  0.659530436  0.101678064 -0.13917718
```

Col comando `summary(pca)` è invece possibile osservare la deviazione standard di ciascuna componente, la proporzione di varianza spiegata, e le proporzioni cumulate.

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5    PC6    PC7
Standard deviation     1.7976 1.0899 1.0130 0.83658 0.64367 0.5493 0.3723
Proportion of Variance 0.4617 0.1697 0.1466 0.09998 0.05919 0.0431 0.0198
Cumulative Proportion  0.4617 0.6313 0.7779 0.87791 0.93710 0.9802 1.0000
```

Rimpiazzando le variabili osservate con le componenti principali è importante osservare come per costruzione esse sono non correlate tra loro (cioè ortogonali) e sono ordinate in ordine decrescente di varianza. Inoltre, la somma delle varianze si conserva nel passaggio dalle variabili osservate alle componenti principali.

Tramite la funzione `screeplot(pca,type=c("lines"))` posso osservare graficamente i fattori e stabilire il numero di componenti principali da tenere in considerazione per rappresentare il fenomeno.



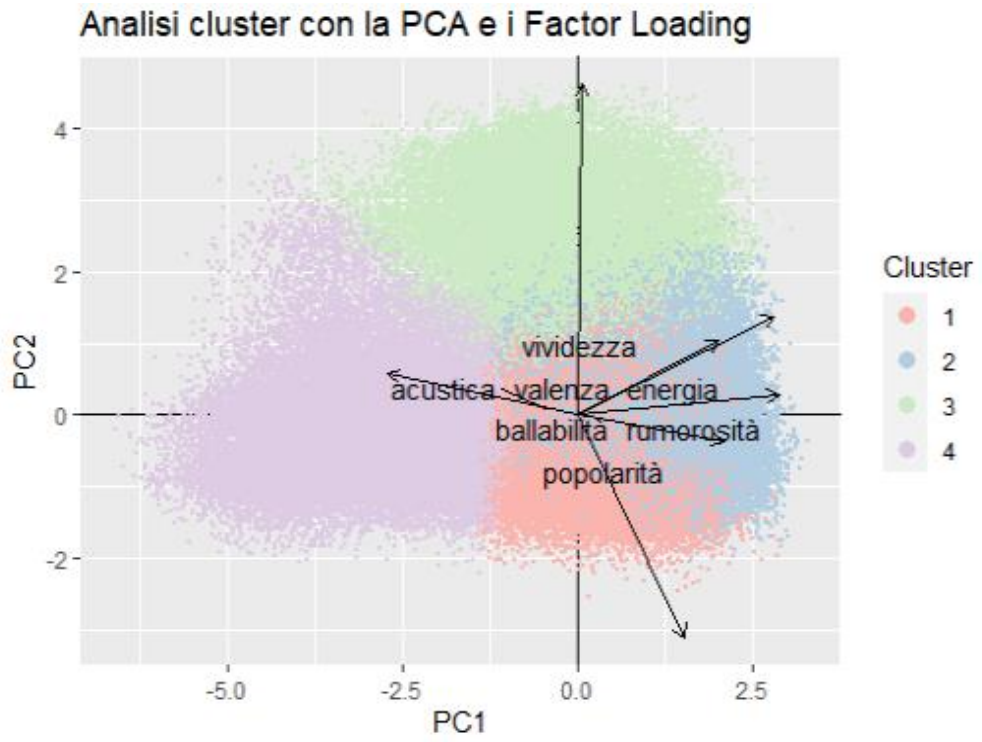
Combino poi i risultati della clusterizzazione con quelli della PCA e coi dati originari.

```
standard <- data.frame(standard,music2[,-c(4:10)])
standard$cluster <- as.factor(kmeans_cluster)
pr1 <- data.frame(pca$x, cluster = factor(kmeans_cluster))
```

Per ottenere una rappresentazione grafica dei risultati dell'analisi PCA in cui venissero rappresentati sia gli score che i factor loading ho sfruttato il *biplot* [3], usufruendo dei pacchetti `ggplot2` e `ggrepel`.

```
library(ggplot2)
library(ggrepel)
dataframe_pc <- data.frame(varnames = rownames(pca$rotation), pca$rotation)
x <- "PC1"
y <- "PC2"
data <- data.frame(obsnames=seq(nrow(pca$x)), pca$x)
mult <- min((max(data[,y]) - min(data[,y])/(max(dataframe_pc[,y])-
min(dataframe_pc[,y]))),(max(data[,x]) - min(data[,x])/(max(dataframe_pc[,x])-
min(dataframe_pc[,x]))))
dataframe_pc <- transform(dataframe_pc,v1 = .9 * mult * (get(x)),v2 = .9 * mult
* (get(y)))
ggplot(pr1, aes(x=PC1, y=PC2)) + geom_hline(aes(yintercept=0), size=.2) +
geom_vline(aes(xintercept=0), size=.2) + coord_equal() +
geom_point(aes(color = cluster),size = 0.2) + geom_segment(data = dataframe_pc,
aes(x=0, y=0, xend=v1, yend=v2), arrow = arrow(length=unit(0.2, "cm"))) +
geom_text_repel(data = dataframe_pc, aes((varnames)),point.padding = -
10,segment.size = 0.5) + scale_color_brewer(palette = "Dark2")+
guides(colour = guide_legend(override.aes = list(size=3)))+
labs(title = "Analisi cluster con la PCA e i Factor Loading",color = "Cluster")
```

[3] Per il biplot mi sono basato sul codice trovato su https://stackoverrun.com/it/q/1654591



Attraverso il biplot è possibile osservare i 4 cluster principali. Inoltre è possibile anche notare le direzioni e delineare le correlazioni inverse che esistono tra le variabili. In particolare si nota come l'acustica ha una forte correlazione negativa le variabili energia e rumorosità, e la vividezza della traccia ascoltata è correlata negativamente alla sua popolarità.

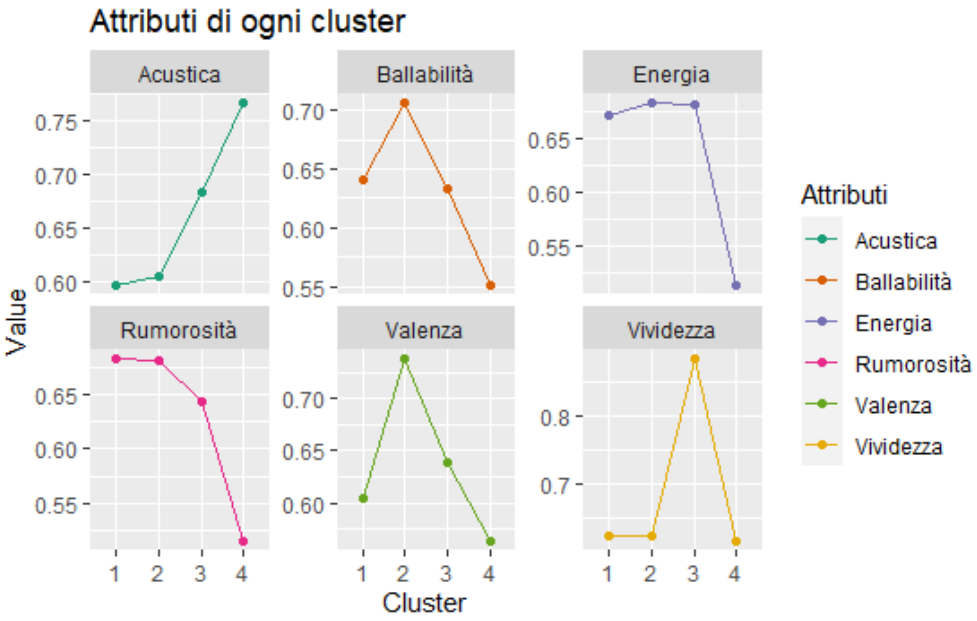
Per analizzare le variabili su ciascuno dei cluster, prima procedo con un ridimensionamento dei dati attraverso la funzione di minimo e massimo per considerare un range da 0 ad 1.

Ho creato quindi un nuovo set di dati a partire da quelli standardizzati, ed ho successivamente utilizzato il comando `cbind`, per reinserire le altre variabili non numeriche.

```
normalize <- function(x){return ((x - min(x))/(max(x) - min(x)))}
standard2 <- normalize(standard[,c(1:7)])
standard2 <- cbind(standard2,standard[,-c(1:7)])
```

Per mostrare graficamente la correlazione tra le variabili che ho considerato e i 4 cluster ho sfruttato il pacchetto `tidyr` e ho riutilizzato `ggplot`.

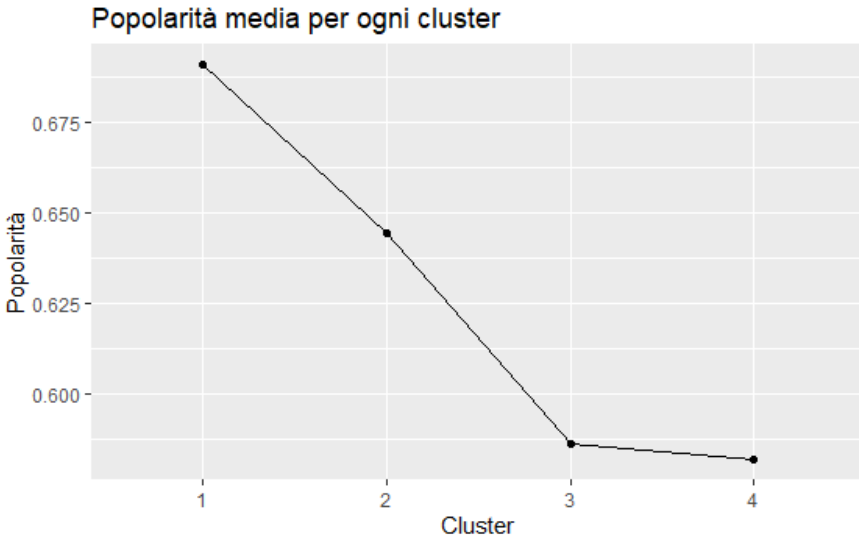
```
library(tidyr)
standard2 %>%
  group_by(cluster) %>%
  summarise(Acustica = mean(acustica),Rumorosità = mean(rumorosità),
            Ballabilità = mean(ballabilità),Energia = mean(energia),
            Vividezza = mean(vividezza),Valenza = mean(valenza)) %>%
  select(Rumorosità,Acustica,Ballabilità,Energia,Vividezza,Valenza,cluster) %>%
  gather("Name","Value",-cluster) %>%
  ggplot(aes(y=Value,x = cluster,col=Name,group = Name)) +
  geom_point()+ geom_line()+ facet_wrap(~Name,scales = "free_y")+
  scale_color_brewer(palette = "Dark2")+
  labs(x="Cluster",col = "Attributi",title = "Attributi di ogni cluster")
```



Dalla rappresentazione grafica è possibile notare che alti valori delle variabili Ballabilità, Energia, Rumorosità e Valenza si collocano comunemente nel Cluster 2, raggiungendo comunque alti valori anche nei Cluster 1 e 3, mentre Vividezza ed Acustica hanno i loro picchi rispettivamente nei Cluster 3 e 4.

Facendo un confronto con la variabile della Popolarità è possibile osservare, invece, che il Cluster 1 risulta essere quello col maggior valore medio di popolarità, mentre il quarto risulta essere quello meno popolare.

```
standard2 %>%

    group_by(cluster) %>%
    summarise(Popolarità = mean(popolarità)) %>%
    select(Popolarità,cluster) %>%
    gather("name","value",-cluster) %>%
    ggplot(aes(y=value,x = cluster,group = name))+
    geom_point()+ geom_line()+
    labs(y = "Popolarità",x="Cluster",title = "Popolarità media per ogni cluster")
```



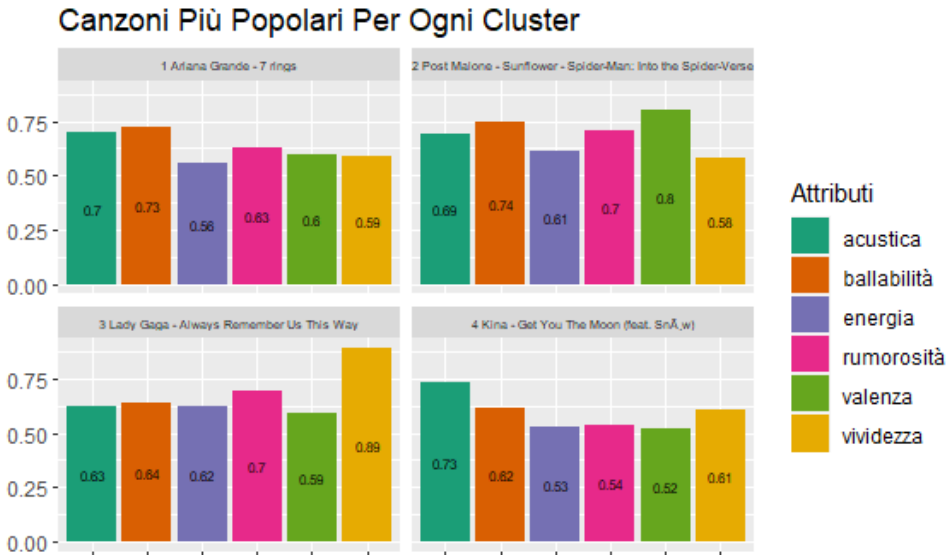
Sfruttando la variabile Popolarità ho poi determinato quali fossero le canzoni più popolari per ciascun cluster.

```
comb = list()
for (i in 1:4){
x <- data.frame(standard2 %>% filter(cluster == i) %>% arrange(desc(popolarità))
%>% head(1))
comb[[i]] <- x}
combine <- do.call(rbind, comb)
combine %>% select(artista,nome_traccia,cluster,popolarità)
```

	artista	nome_traccia	cluster	popolarità
1	Ariana Grande	7 rings	1	0.9349537
2	Post Malone	Sunflower - Spider-Man: Into the Spider-Verse	2	0.9201077
3	Lady Gaga	Always Remember Us This Way	3	0.8755697
4	Kina	Get You The Moon (feat. SnÃ¸w)	4	0.8359804

Infine ho rappresentato graficamente il risultato ottenuto.

```
combine %>%
gather("name","value",2:7) %>%
mutate(label = as.character(paste(cluster,artista,"-", (nome_traccia))), text =
round(value,2)) %>%
arrange(artista) %>%
ggplot(aes(x=name,y=value,fill = (name)))+
geom_col(position = position_stack(),aes(fill = (name)))+
geom_text(aes(label=text),position = position_stack(vjust = .5),size=2)+
facet_wrap(~label)+
scale_fill_brewer(palette = "Dark2")+
theme(axis.text.x = element_text(size = 0),strip.text = element_text(size = 5))+
labs(x=NULL,y =NULL,fill = "Attributi",title = "Canzoni Più Popolari Per Ogni Cl
uster")
```



Conclusioni

Dall'analisi ho potuto osservare le differenze tra i diversi cluster, evidenziando come in particolare nel primo e nel secondo (che risultano essere anche i più popolari ed includono quindi la musica più ascoltata) vadano a confluire alti valori nelle variabili Ballabilità, Energia, Rumorosità e Valenza. Queste variabili risultano infatti essere correlate negativamente all'Acustica e alla Vividezza, che registrano i valori più alti nei due cluster rimanenti.