Tesina Cluster

Alessio Splendiani

May 2024

1 Capitolo 1

1.1 Introduzione

Il **problema** che vogliamo affrontare è il seguente: ***HELP International***(è una ONG umanitaria internazionale impegnata a combattere la povertà e a fornire alle popolazioni dei paesi arretrati servizi di base e aiuti durante i periodi di disastri e calamità naturali.) La ONG è riuscita a raccogliere circa 10 milioni di dollari.

Ora l'amministratore delegato della ONG deve decidere come utilizzare questo denaro in modo strategico ed efficace. Pertanto, il CEO deve prendere la decisione di scegliere i paesi che hanno più disperato bisogno di aiuti. Il mio lavoro come Data Scientist è quello di classificare i paesi utilizzando alcuni fattori socio-economici e sanitari che determinano lo sviluppo complessivo del paese, così da suggerire al CEO i paesi su cui bisogna concentrarsi maggiormente.

Il Dataset che stiamo utilizzando è "Country Data" ed è stato preso da (https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data/data), comprende 167 paesi osservati su 10 colonne.

country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
Afghanistan	90.2	10	7.58	44.9	1610	9.44	56.2	5.82	553
Albania	16.6	28	6.55	48.6	9930	4.49	76.3	1.65	4090
Algeria	27.3	38.4	4.17	31.4	12900	16.1	76.5	2.89	4460
Angola	119	62.3	2.85	42.9	5900	22.4	60.1	6.16	3530
Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200
Argentina	14.5	18.9	8.1	16	18700	20.9	75.8	2.37	10300
Armenia	18.1	20.8	4.4	45.3	6700	7.77	73.3	1.69	3220
Australia	4.8	19.8	8.73	20.9	41400	1.16	82	1.93	51900
Austria	4.3	51.3	11	47.8	43200	0.873	80.5	1.44	46900
Azerbaijan	39.2	54.3	5.88	20.7	16000	13.8	69.1	1.92	5840
Bahamas	13.8	35	7.89	43.7	22900	-0.393	73.8	1.86	28000
Bahrain	8.6	69.5	4.97	50.9	41100	7.44	76	2.16	20700
Bangladesh	49.4	16	3.52	21.8	2440	7.14	70.4	2.33	758
Barbados	14.2	39.5	7.97	48.7	15300	0.321	76.7	1.78	16000
Belarus	5.5	51.4	5.61	64.5	16200	15.1	70.4	1.49	6030

Figure 1: Country Data

Andiamo a fare una breve descrizione delle variabili:

1. **Country**: Indica il nome del Paesi;
2. **Child-mort**: Indica il numero di bambini morti sotto i 5 anni ogni 1000 nati vivi;
3. **Exports**: Fa riferimento alle esportazioni di beni e servizi pro capite, espresso in percentuale del PIL pro capite;
4. **Health**: Indica la spesa sanitaria totale pro capite, espresso in percentuale del PIL pro capite;
5. **Imports**: Indica le Importazioni di beni e servizi pro capite, espresso in percentuale del PIL pro capite;

6. **Income**: Si riferisce al reddito netto pro capite;
7. **Inflation**: Aumento prolungato del livello medio generale dei prezzi di beni e servizi;
8. **Life-expec**: Il numero medio di anni che un neonato vivrebbe se gli attuali modelli di mortalità rimanessero gli stessi;
9. **Total-fer**: Il numero di bambini che nascerebbero da ciascuna donna se gli attuali tassi di fertilità per età rimanessero gli stessi;
10. **Gdpp**: Il PIL pro capite.Calcolato come il PIL totale diviso per la popolazione totale.

2 Capitolo 2: Cluster dei paesi

2.1 Analisi esplorativa

Inziamo con l'analisi del Dataset andando a fare prima un'analisi esplorativa e poi andando ad applicare i diversi algoritmi descritti precedentemente. Dopo aver verificato che nel nostro DataSet non sono presenti valori mancanti siamo andati ad osservare eventuali correlazioni tra le variabili.

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
child_mort	1.0000000	-0.3180986	-0.20040206	-0.12720751	-0.5243150	0.2882751	-0.88667610	0.8484781	-0.4830322
exports	-0.3180986	1.0000000	-0.11441838	0.73738130	0.5167834	-0.1073043	0.31631644	-0.3200180	0.4187246
health	-0.2004021	-0.1144184	1.00000000	0.09570328	0.1295786	-0.2553718	0.21069212	-0.1966740	0.3459655
imports	-0.1272075	0.7373813	0.09570328	1.00000000	0.1224023	-0.2470117	0.05438784	-0.1590541	0.1154945
income	-0.5243150	0.5167834	0.12957861	0.12240235	1.0000000	-0.1477593	0.61196247	-0.5018401	0.8955714
inflation	0.2882751	-0.1073043	-0.25537176	-0.24701172	-0.1477593	1.0000000	-0.23970699	0.3169209	-0.2216294
life_expec	-0.8866761	0.3163164	0.21069212	0.05438784	0.6119625	-0.2397070	1.00000000	-0.7608747	0.6000891
total_fer	0.8484781	-0.3200180	-0.19667399	-0.15905406	-0.5018401	0.3169209	-0.76087469	1.0000000	-0.4549103
gdpp	-0.4830322	0.4187246	0.34596553	0.11549452	0.8955714	-0.2216294	0.60008913	-0.4549103	1.0000000

Figure 2: Matrice di Correlazione

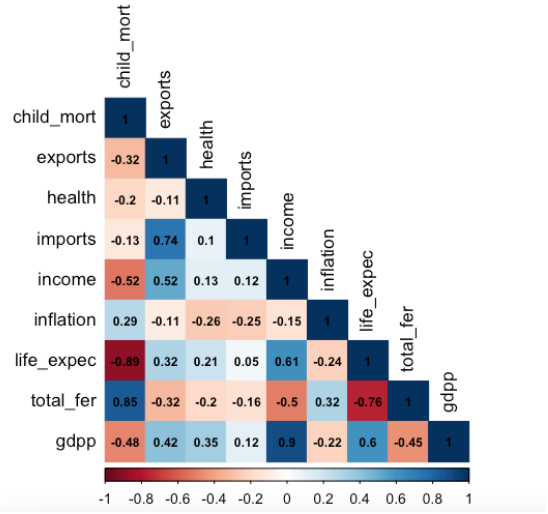

Figure 3: Corrplot delle variabili

Si può subito notare la presenza di correlazioni tra variabili:

- **child-mort** aumenta quando il reddito, il GDP e le esportazioni diminuiscono;
- **exports** aumenta all'aumentare delle importazioni, del reddito e del GDP;
- **life-expec** aumenta all'aumentare del reddito e del GDP;
- **income** è estremamente correlato con il GDP;

- **inflation** un'inflazione elevata mostra un total-fer e un child-mort elevati. Questo desrcive le caratteristiche tipiche dei paesi più arretrati.

Successivamente sono ad andato ad osservare i paesi migliori e peggiori per ogni variabie considerata.

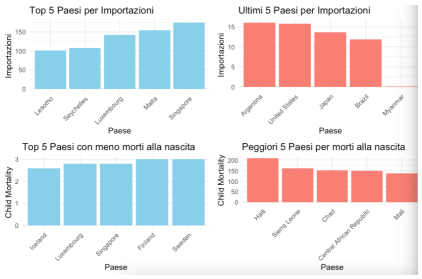

Figure 4: Importazioni e morti alla nascita

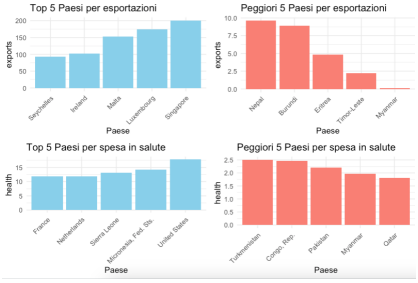

Figure 5: Esportazioni e Sanità

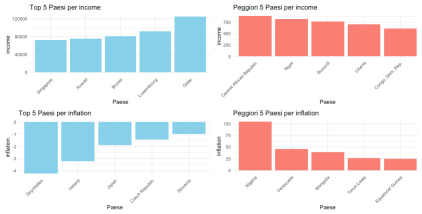

Figure 6: Reddito e inflazione

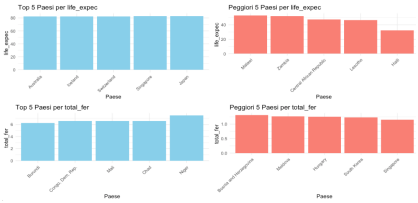

Figure 7: Aspettativa di vita e fertilità

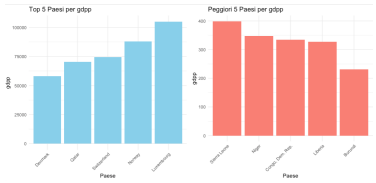

Figure 8: GDP

Attraverso i grafici a barre mostrati possiamo ricavare diverse informazioni. Per esempio è possibile vedere come la **Svizzera** sia la migliore sia per l'aspettativa di vita che per il GDP. Paesi come Singapore, Malta, Lussemburgo e Seychelles sono presenti nella top 5 delle esportazioni e delle importazioni I paesi dell'**Africa** sono tra i peggiori per Aspettativa di vita, spesa in sanità e Income. Inoltre sono anche quelli com il maggior valore di inflazione.

2.2 PCA

La PCA (Principal Component Analysis) è una tecnica di riduzione della dimensionalità che viene utilizzata per semplificare dataset complessi con molte variabili intercorrelate. L'obiettivo della PCA è trasformare le variabili originali in un nuovo insieme di variabili, chiamate componenti principali, così da catturare la maggior parte della variabilità presente nei dati. La prima componente principale cattura la massima varianza possibile, la seconda componente principale cattura la massima varianza possibile residua, e così via. Questo permette di ridurre il numero di variabili mantenendo gran parte dell'informazione originale, semplificando l'analisi e la visualizzazione dei dati.

Inizialmente l'idea era quella di fare un'analisi delle componenti principali sulle variabili scalate del Dataset, solo che andando a selezionare solo le prime 2 componenti principali la varianza spiegata era solo del 63 %

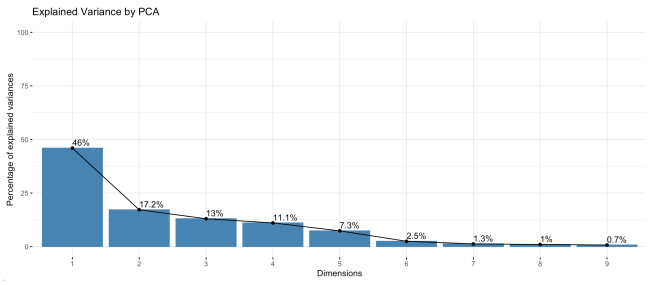

Figure 9: Varianza spiegata

Quindi o si andavano a selezionare le prime 3 componenti principali, oppure, si proseguiva con delle tecniche di feature Engineering. Abbiamo scelto di adottare la seconda opzione, questa tecnica consiste nella creazione di nuove variabili a partire dai dati originali. L'obiettivo è quello di migliorare le prestazioni dei modelli.

```
# Raggruppo le variabili in base alle categorie
Country_data$health_index <- rowMeans(Country_data[, c("child_mort"
    , "health", "life_expec", "total_fer")])
Country_data$trade_index <- rowMeans(Country_data[, c("imports", "
    exports")])
Country_data$finance_index <- rowMeans(Country_data[, c("income", "
    inflation", "gdpp")])

# Normalizza le nuove caratteristiche aggregate
Country_data_normalized <- scale(Country_data[, c("health_index", "
    trade_index", "finance_index")])
```

Abbiamo raggruppato le variabili in tre nuove categorie di variabili (**health-index, trade-index, finance-index**)in base anche alla correlazione che le variabili avevano tra di loro. Facendo ora le componenti principali su questo gruppo di variabili è possibile vedere come la varianza spiegata dalle prime 2 componenti principali sia dell'83 %

```r
    #Analisi delle Componenti principali
pca_result <- prcomp(Country_data_normalized, scale. = TRUE)
fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 100) , main = "
    Explained Variance by PCA")
# Visualizza i loadings per la PC1
pc1_loadings <- pca_result$rotation[, 1]
print(pc1_loadings)
# Visualizza i loadings per la PC2
pc2_loadings <- pca_result$rotation[, 2]
print(pc2_loadings)
```

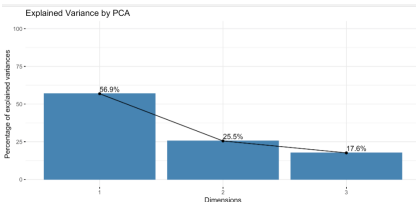

Figure 10: Varianza spiegata

```
> # Visualizza i loadings per la PC1
> pc1_loadings <- pca_result$rotation[, 1]
> print(pc1_loadings)
 health_index   trade_index finance_index
   -0.5901318     0.5092635     0.6264145
> # Visualizza i loadings per la PC2
> pc2_loadings <- pca_result$rotation[, 2]
> print(pc2_loadings)
 health_index   trade_index finance_index
    0.5092064     0.8369190    -0.2006873
```

Figure 11: Loadings PC1 e PC2

La **prima componente principale** spiega il 56.9 % della varianza totale e sembrerebbe essere correlata positivamente all' indice finanziario che include (income, inflation, gdp) e all'indice del commercio che include (imports, exports) al contrario è negativamente correlato all'indice sulla salute che include (child-mort, health,life-expec,total-fer).

La **seconda componente principale** spiega il 25.5 % della varianza totale e sembrerebbe essere correlata positivamente con l'indice del commercio e l'indice sulla salute.

Si è poi deciso di effettuare dei test attraverso degli indici come *Elbow*, *Silhouette* ed *NbClust* così da scegliere il numero ottimale di cluster per la nostra analisi.

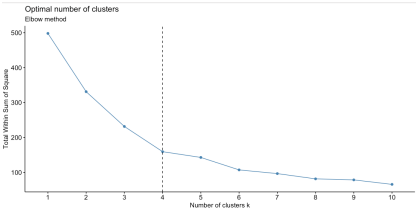

Figure 12: Indice di Elbow

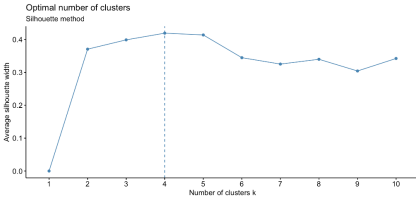

Figure 13: Indice di Silhouette

```
*******************************************************************
* Among all indices:
* 2 proposed 2 as the best number of clusters
* 7 proposed 3 as the best number of clusters
* 6 proposed 4 as the best number of clusters
* 2 proposed 5 as the best number of clusters
* 1 proposed 7 as the best number of clusters
* 1 proposed 9 as the best number of clusters
* 2 proposed 10 as the best number of clusters
* 2 proposed 15 as the best number of clusters

                ***** Conclusion *****

* According to the majority rule, the best number of clusters is  3
```

Figure 14: NbClust

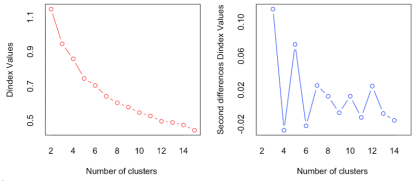

Figure 15: Output NbClust

Elbow Method: il metodo del gomito consiste nel calcolare il valore della somma dei quadrati delle distanze tra i punti di dati e i centroidi dei cluster. Questo valore viene calcolato per diversi valori di k (il numero di cluster) e viene tracciato in un grafico. L'idea è individuare il punto in cui la somma dei quadrati delle distanze inizia a diminuire più lentamente, creando un "gomito" nel grafico. Questo punto suggerisce il numero ottimale di cluster da utilizzare per il clustering dei dati.

Indice di Silhouette: Questo indice misura quanto ogni punto dei dati è simile ai punti del suo stesso cluster rispetto ai punti degli altri cluster. Varia da -1 a 1, dove valori più alti indicano cluster più coesi.

L'indice NbClust include diversi indici tra cui anche Elbow, Silhouette, Dunn e altri, con l'obiettivo di identificare il numero di cluster che massimizza la coesione all'interno dei cluster e la separazione tra di essi. In base a questo indice abbiamo scelto di selezionare $\mathbf{K = 3}$. Il grafico a sinistra mostra i valori di Dindex per ogni numero di cluster.

Il Dindex è un indice di clustering che misura la compattezza e la separazione dei cluster. Il valore che indica una diminuzione più marcata potrebbe suggerire il numero ottimale di cluster. Questo indica il valore che ha una migliore coesione all'interno dei cluster e una migliore separazione tra di essi.

Il grafico a destra mostra le differenze seconde dei valori di Dindex per ogni numero di cluster. Le differenze seconde sono utilizzate per identificare picchi significativi nel grafico di Dindex, che possono suggerire un numero ottimale di cluster.

2.3 K-MEANS

K-means è un algoritmo di clustering utilizzato per dividere un insieme di dati in k gruppi distinti. L'algoritmo inizia scegliendo k punti iniziali come **centroidi**, che rappresentano il centro di ciascun gruppo. Ogni punto nel dataset viene quindi assegnato al centroide più vicino, formando i cluster iniziali. Dopo questa assegnazione, i centroidi vengono aggiornati calcolando la media dei punti appartenenti a ciascun cluster. Il processo di assegnazione e aggiornamento continua fino a quando i centroidi si stabilizzano, ovvero le assegnazioni dei punti non cambiano più.

Tra i punti di **forza** del K-means abbiamo:

- soluzione efficiente dal punto di vista computazionale;
- popolare;
- intuitivo.

Tra i punti di **debolezza**:

- Spesso termina in un ottimo locale;
- È necessario specificare k, il numero di cluster, in anticipo (ci sono modi per determinare automaticamente il migliore k;
- Influenzato da errori di misura ed outlier.

```
#Algoritmo K-Means
clcountry <- kmeans(Country_data_normalized,centers = 3)
table(clcountry$cluster)
#Utilizziamo le prime 2 componenti principali
kk <- prcomp(Country_data_normalized,2)$x[,1:2]
Country_data$country<- as.character(Country_data$country)
par(mfrow=c(1,1),mar=c(4.5,4.5,1,1))
plot(kk,cex=1,pch=19,col=c("red","blue","green")[clcountry$cluster
    ])
text(kk,Country_data$country,pos=1,col=c("red","blue","green")[
    clcountry$cluster],cex=0.8)
```

Attraverso queste righe di codice andiamo ad utilizzare il K-Means fissando K = 3 come consigliato dagli indici visti precedentemente, ed abbiamo fatto un plot bidimensionale dei vari paesi dove sulle x abbiamo la PC1 e sulle y la PC2.

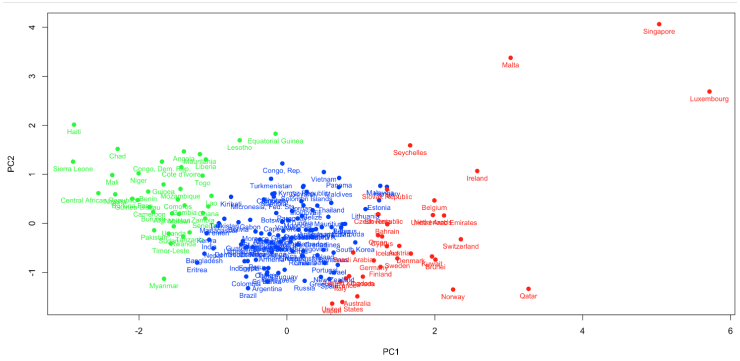

Figure 16: Plot dei paesi utilizzando K-MEANS

Osservando il grafico è possibile fare delle osservazioni:

- **Cluster 1:** Paesi (37) che si trovano sulla destra del grafico, hanno alte importazioni ed esportazioni (trade-index) e godono di una situazione finanziaria buona (finance-index). Troviamo la maggior parte dei Paesi Europei.

- **Cluster 2:** Paesi (97) al centro che hanno valori medi sia per importazioni ed esportazioni (trade-index) ed una situazione finanziaria nella media (finance -index). Troviamo paesi come Brasile, Argentina, Indonesia ecc...
- **Cluster 3**: Paesi (33) sulla sinistra del grafico, caratterizzati da basse importazioni ed esportazioni (trade-index) e una bassa spesa in sanità rispeto al gdp, bassa aspettativa di vita e alta mortalità infantile (health-index). Troviamo la maggior parte dei Paesi dell'Africa.

Infine è possibile osservare come ci siano degli outlier: Lussemburgo, Malta e Singapore che hanno alti valori di trade-index, finance-index e healt-index, potrebbero essere messi in un cluster a parte selezionando K = 4. Sul codice R è stata analizzata anche questa situazione.

2.4 Algoritmo PAM

PAM (Partitioning Around Medoids) è un algoritmo di clustering che divide i dati in k gruppi, simile a K-means ma più robusto ai valori anomali. Inizia da un set iniziale di **medoidi** (l'unità situata più centralmente in un gruppo) e sostituisce iterativamente uno dei medoidi con uno dei non-medoidi se migliora la distanza totale del clustering risultante. PAM funziona efficacemente per piccoli data-set, ma non si adatta bene ai grandi insiemi di dati (a causa della complessità computazionale)

- **PAM** è più robusto di k-means in presenza di errori di misura e valori anomali perché un medoide è meno influenzato da valori anomali o altri valori estremi di quanto lo sia la media.
- **PAM** funziona in modo efficiente per piccoli set di dati, ma non si adatta bene ai grandi insiemi di dati.

```
par(mar=c(5,5,1,1))
fviz_cluster(clcountry,
             data = Country_data_normalized,
             geom = "point",
             main ="K-MEANS",
             labelsize = 3,
             show.clust.cent = TRUE,
             ggtheme = theme_minimal()) +
  geom_text(aes(label = Country_data$country), size = 3, vjust =
      -1)
pam_result <- pam(Country_data_normalized,3)
# Visualizza i risultati del clustering con PAM
fviz_cluster(pam_result,
             data = Country_data_normalized,
             geom = "point",
             main ="PAM",
             labelsize = 3,
             show.clust.cent = TRUE,
             ggtheme = theme_minimal()) +
  geom_text(aes(label = Country_data$country), size = 3, vjust =
      -1)

table(clcountry$cluster)
table(pam_result$cluster)
```

Attraverso il comando table possiamo osservare come i paesi siano stati clusterizzati in base all'algoritmo PAM. Nel K-Means avevamo 33 Paesi nel cluster 1, 97 nel cluster 2 e 37 nel cluster 3.

```
> table(pam_result$cluster)

 1  2  3
47 82 38
```

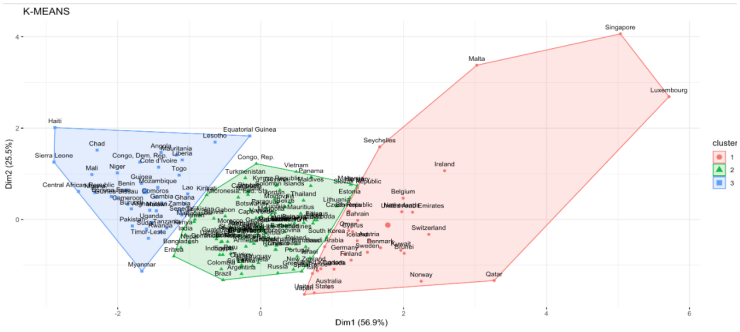

Figure 17: Plot dei paesi utilizzando K-MEANS

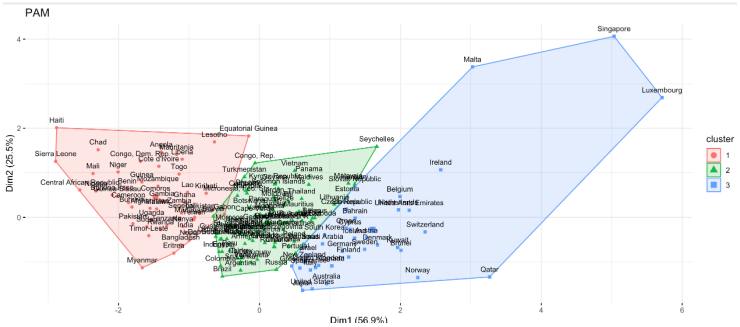

Figure 18: Plot dei paesi utilizzando PAM

Osservando il plot possiamo notare come tra i due algoritmi ci siano differenze nella clusterizzazione di alcuni paesi, ad esempio Bangladesh, Eritrea, India nel K-Means vengono inseriti nel primo cluster, utilizzando il PAM vengono inseriti nel 2. Anche le Seychelles nel plot del K-Means vengono inserite nel 3 cluster, con il PAM, invece, nel 2.

2.5 Clustering gerarchico

I metodi di clustering gerarchico si basano sulla matrice delle distanze per definire dei criteri di raggruppamento, sono algoritmi utilizzati per raggruppare dati in una struttura ad albero, chiamata dendogramma, che rappresenta la gerarchia delle relazioni tra i dati. Esistono due principali approcci nei metodi gerarchici: **agglomerativo** (DIANA) e **divisivo** (AGNES). I metodi agglomerativi partono dalle single unità in n e raggruppano fino al cluster singolo. Al contrario, i divisivi partono dal gruppo contenente tutte le unità disponibili e procedono a dividere tale gruppo in clusters sempre più piccoli fino alle singole unità.

Un raggruppamento delle unità si può ottenere "tagliando" il dendrogramma ad un livello desiderato, così che, quindi, ciascuna componente connessa ad un

nodo forma un Gruppo. Al fine di determinare il taglio nel dendrogramma e di definire dei clusters tramite i metodi suggeriti è utile definire delle misure di distanza fra i gruppi.

Nel DataSet è stato utilizzato il **Metodo di Ward** secondo tale approccio utilizzato per la costruzione in logica agglomerativa, la similarità di due gruppi si basa sull'incremento di SSE (Sum of Squared Errors) quando i due gruppi vengono uniti:

- tanto minore l'incremento tanto più elevata la similarità:

```r
#Clustering gerarchico
# Calcola la matrice di distanza
dist_matrix <- dist(Country_data_normalized, method = "euclidean")

# Esegui il clustering gerarchico usando il metodo agglomerativo
hc <- hclust(dist_matrix, method = "ward.D2")

# Visualizza il dendrogramma
plot(hc, labels = Country_data$country, main = "Dendrogramma del
    Clustering Gerarchico", cex = 0.8)
rect.hclust(hc, k = 3, border = 2:5) # Taglia il dendrogramma in 3
    cluster
```

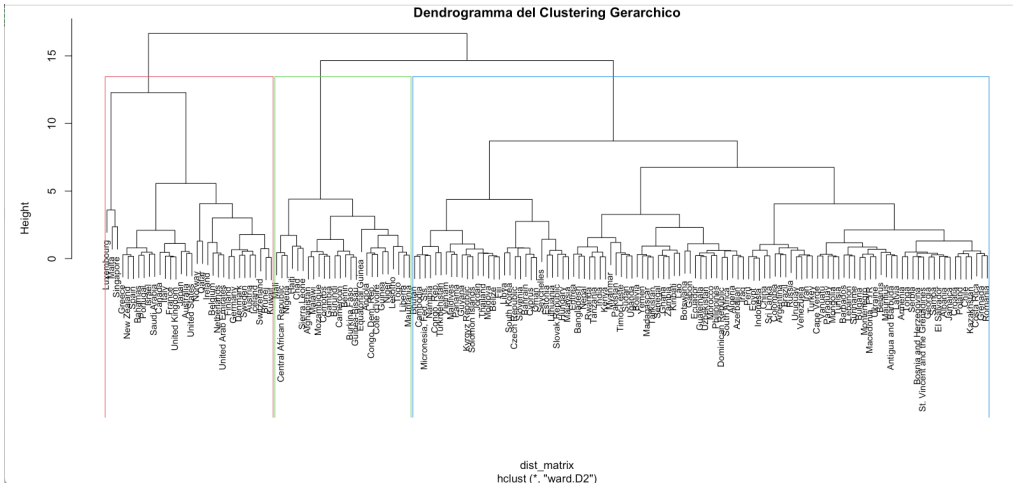

Figure 19: Dendrogramma

I cluster risultanti dal dendogramma sembrano distinguersi principalmente per le caratteristiche socio-economiche delle nazioni rispettando quanto visto precedentemente con gli altri algoritmi. Il **cluster sulla sinistra** è caratterizzato da nazioni Europee e da Malta, Lussemburgo e Singapore che abbiamo visto hanno un alto finance-index e un alto trade-index. Il **cluster centrale** è composto principalmente da nazioni dell'Africa che hanno un basso finance-index e una basso health-index. Infine, il **terzo cluster sulla destra** include principalmente nazioni che hanno valori nella media per tutte e tre le variabili.

2.6 Clustering basati sulle distribuzioni: DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) è un algoritmo che si basa su una nozione di gruppo basata sulla densità: un gruppo è definito come un insieme massimo di punti connessi alla densità. Individua i gruppi di forma arbitraria nei database spaziali contenenti molto rumore. Utilizza due parametri: Eps (raggio massimo del vicinato) e MinPts (numero minimo di punti nello Eps-neighbourhood di quella unità)

Algoritmo **DBSCAN**:

1. Arbitrariamente seleziona una unità p;
2. Recupera tutti i punti con densità raggiungibile da p tenuto conto di Eps e MinPts;
3. Se p è un'unità centrale,viene formato un gruppo;
4. Se p è un'unità di confine, nessuna unità è raggiungibile in densità da p e DBSCAN visita l'unità successiva del database;
5. Continua il processo fino a quando tutte le unità sono state elaborate.

```r
# Calcola il clustering
dbscan_result <- dbscan(Country_data_normalized, eps = 0.45, minPts = 8)

# Andiamo ad associare i colori ai nostri cluster
cluster_colors <- c("red", "green", "blue")  # Colori per i cluster
noise_color <- "black"  # Colore per i punti di rumore

# Visualizza i risultati del clustering DBSCAN
plot(kk, col = dbscan_result$cluster + 1, pch = 19, main = "DBSCAN Clustering")
text(kk, Country_data$country, pos = 1, col = dbscan_result$cluster + 1, cex = 0.7)
```

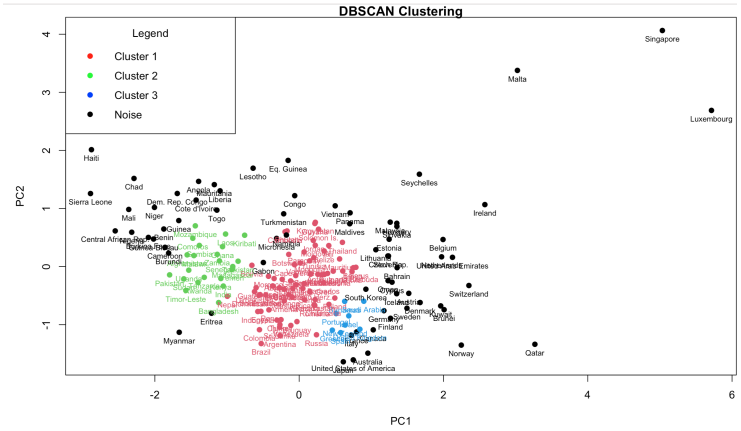

Figure 20: Plot con DBSCAN

Il plot dell'algoritmo DBSCAN mostra una chiara suddivisione dei dati in tre cluster principali, rappresentati dai colori rosso, verde e blu. Inoltre, ci sono numerosi punti "Noise" (67), che non sono stati assegnati a nessun cluster, poiché non soddisfano i requisiti di densità definiti dai valori scelti per eps e minPts.

- Primo cluster (rosso): Questo è il cluster più grande, comprendente 70 paesi. Questi paesi presentano valori medi nei tre indici considerati: trade-index, finance-index e health-index. Si trovano in una situazione intermedia per quanto riguarda le variabili analizzate.
- Secondo cluster (verde): Questo cluster include 22 paesi, principalmente l'India e diversi paesi africani. Come evidenziato anche dagli altri algoritmi di clustering, questi paesi mostrano i valori più bassi in tutti gli indici considerati, evidenziando situazioni critiche nei settori del commercio, economici e della salute.
- Terzo cluster (blu): Questo cluster comprende 8 paesi europei. Questi paesi si distinguono per avere buoni valori sia nel finance-index che nel trade-index, indicando una situazione finanziaria buona.

La significativa presenza di punti "Noise" indica che una parte consistente dei dati non si adatta ai modelli di densità stabiliti. Questo potrebbe riflettere la varietà dei paesi analizzati, suggerendo che esistono gruppi con profili unici che non si adattano facilmente ai cluster identificati. Potrebbe essere utile modificare i parametri di eps e minPts per migliorare ulteriormente la suddivisione dei dati.

3 Capitolo 3

3.1 Conclusioni

L'obiettivo di questo studio era identificare i Paesi che necessitano maggiormente di aiuto da parte della ONG (Help International) al verificarsi di periodi di disastri e calamità naturali, utilizzando vari algoritmi di clustering per analizzare i dati sull'economia, sul commercio e sulla salute. I metodi utilizzati includono K-means, PAM (Partitioning Around Medoids), Clustering gerarchico e DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

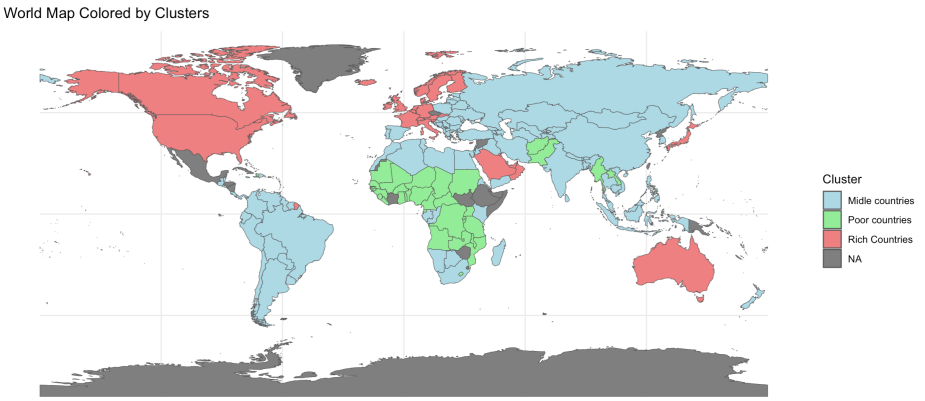

Figure 21: Mappa del mondo

I vari algoritmi di clustering utilizzati in questo studio hanno mostrato una convergenza nell'identificare i Paesi che necessitano maggiormente di aiuto. In particolare, i Paesi situati in Africa Centrale e Asia Meridionale (Afghanistan, Pakistan e Myanmar), sono emersi costantemente come quelli con i punteggi più bassi negli indici economici, di commercio e salute.

Indipendentemente dall'algoritmo di clustering utilizzato (K-means, PAM, clustering gerarchico o DBSCAN), i Paesi contrassegnati in verde sono stati identificati come quelli che necessitano di maggiori aiuti. Questa coerenza tra i diversi metodi di clustering rafforza la validità delle conclusioni ottenute.

Alla luce di questi risultati, si raccomanda alla ONG (HELP International), in caso di calamità naturali o periodi di difficoltà, di concentrare i propri sforzi e risorse su questi Paesi identificati come i più bisognosi. Intervenire in queste aree potrebbe portare a un miglioramento significativo delle loro condizioni socio-economiche e sanitarie, massimizzando l'impatto positivo degli aiuti forniti.