GeoCluster for HELP International

Repository: github.com/ocone28/ClusteringCountriesFIA

SOMMARIO:

INTRODUZIONE

HELP International è una ONG umanitaria internazionale che si impegna a combattere la povertà e fornire alle popolazioni dei paesi arretrati servizi di base e soccorso durante il periodo di disastri e calamità naturali.

L'obbiettivo è quello di classificare i paesi utilizzando fattori socio-economici e sanitari che determinano lo sviluppo complessivo del paese.

GeoCluster for HELP International

DEFINIZIONE DELL' AMBIENTE

La prima fase dell'analisi non poteva che essere lo studio e la descrizione dell'ambiente.
Per fare ciò si è deciso di usare il modello PEAS (Performance, Environment, Actuators, Sensors) per schematizzare quelle che sono le sue caratteristiche principali:

Performance: Le prestazioni saranno misurate in funzione dell'accuratezza con cui l'agente riuscirà a collocare le diverse Nazioni all'interno dei cluster.
Environment: L'ambiente è a singolo agente, ed è completamente osservabile, dato che i sensori dell' ambiente danno accesso allo stato completo dell' ambiente in qualsiasi momento.
Actuators: Gli attuatori dell'agente consistono nel mettere a disposizione n gruppi, contenenti le Nazioni presenti all interno del dataset.
Sensors: I sensori dell'agente consistono nel mettere a disposizione un dataset con numerosi campi che fungeranno da sensori per il problema preposto.

GeoCluster for HELP International

SCELTA DEL DATASET

Ci serviamo di un dataset costituito con i seguenti campi.

- **Id** Identificativo univoco Nazione
- **Country** Nome della Nazione
- **Child_mort** Morte di bambini sotto i 5 anni ogni 1000 nati vivi
- **Exports** Esportazioni di beni e servizi pro capite. Espresso in percentuale del PIL pro capite
- **Health** Spesa sanitaria totale pro capite. Espresso in percentuale del PIL pro capite
- **Imports** Importazioni di beni e servizi pro capite. Espresso in percentuale del PIL pro capite
- **Income** Reddito netto pro capite
- **Inflation** La misura del tasso di crescita annuo del PIL totale
- **Life_expec** Il numero medio di anni che un neonato vivrebbe se gli attuali modelli di mortalità dovessero rimanere gli stessi
- **Total_fer** Il numero di bambini che nascerebbero da ciascuna donna se gli attuali tassi di età e fertilità rimanessero gli stessi.
- **Gdpp** Il PIL pro capite. Calcolato come il PIL totale diviso per la popolazione

id	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	45.5	6.03	58.9	19100	1.44	76.8	2.13	12200

GeoCluster for HELP International

TIPOLOGIA APPRENDIMENTO

Nello sviluppo del progetto si è deciso di implementare due algoritmi di apprendimento non supervisionato, in particolare il K-MEANS e il DBSCAN.

Il DBSCAN è stato presto scartato poiché si è visto avere un coefficiente di forma molto più alto rispetto al K-MEANS:
- coefficiente di forma DBSCAN: 0.536
- coefficiente di forma K-MEANS: 0.224

Quindi si è tenuto conto solo dello sviluppo del K-MEANS.

Il K-MEANS ha come obiettivo creare cluster (gruppi) di campioni che abbiano caratteristiche simili partendo da un dataset iniziale di addestramento, non etichettato.

Il K-MEANS ha una proprietà fondamentale: il numero di cluster è definito in anticipo.

GeoCluster for HELP International

IMPLEMENTAZIONE PT.1

Il primo step implementativo è stato quello di pulire il dataset.

1. Abbiamo eliminato le colonne testuali che influivano negativamente sul trattamento dei dati, quindi abbiamo eliminato la colonna 'Country' e abbiamo rimosso anche le caratteristiche non discrete come 'id'.

2. Lo step successivo è stato quello di pulire i dati da tuple contenenti valori nulli (NaN) che rendevano difficile il trattamento degli stessi.

Ora il dataset è pronto per poter essere processato.

```
#togliamo le colonne contenenti dati testuali per facilitare il calcolo
stringless_df = df.drop(['id','country'], axis=1)
#togliamo anche le tuple che presentano valori NaN
stringless_df = stringless_df.dropna()
stringless_df.head(5)
```

GeoCluster for HELP International

IMPLEMENTAZIONE PT.2

Successivamente a causa della presenza di più caratteristiche discrete abbiamo utilizzato l'analisi delle componenti principali (PCA) per poter estrarre tutte le caratteristiche non concorrenti e quindi riducendo la complessità del problema passando da uno spazio di rappresentazione più consono.

```
#Caliamo i dati per una migliore distribuzione
X_std = StandardScaler().fit_transform(stringless_df)
pca = PCA(n_components=.95)
principalComponents = pca.fit_transform(X_std)

#Applichiamo l'analisi delle componeneti principali
pca = PCA(n_components=9)
principalComponents = pca.fit_transform(X_std)
#abbiamo a questo punto un nuovo dataframe dopo l'esecuzione di PCA
pca_df = pd.DataFrame(principalComponents)
```

0	1	2	3	4	5	6	7	8
2.960176	-0.387071	-0.963682	0.806715	-0.133060	-0.198175	0.474683	0.411509	0.123987
-0.513631	0.057112	-0.423658	-1.292049	0.117514	0.128391	0.295438	-0.163085	-0.217092
0.144422	-0.591171	1.135052	-0.974524	0.161719	-0.401101	-0.051276	-0.139424	-0.144734
2.938836	0.620269	2.014763	1.333221	-0.196226	-0.626685	-0.450536	-0.412543	0.005579
-1.119407	0.795362	-0.013968	-0.624320	-0.218793	-0.180023	0.337748	-0.026815	-0.095145

GeoCluster for HELP International

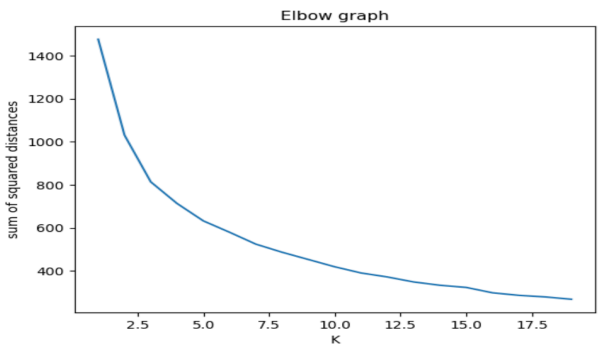
IMPLEMENTAZIONE PT.3

I dati, finita la fase di scaling con PCA, sono stati dati in input alla funzione di fitness del K-MEANS, la quale calcola la distanza di ogni dato dall'altro secondo la distanza euclidea.

La scelta di K è avvenuta con il metodo del gomito tramite il calcolo della somma dei quadrati degli errori, di cui sotto il risultato.

```
#calcola la somma degli errori quadrati per valutare
#il valore di k da utilizzare per il kmeans
#utilizziamo il metodo del gomito per determinare il valore di k
sum_of_squared_distances = []
K = range(1,20)
for k in K:
    km = KMeans(n_clusters=k)
    km = km.fit(pca_df)
    sum_of_squared_distances.append(km.inertia_)

ax = sns.lineplot(x=K, y = sum_of_squared_distances)
ax.set(xlabel='K', ylabel='sum of squared distances', title='Elbow graph')
```



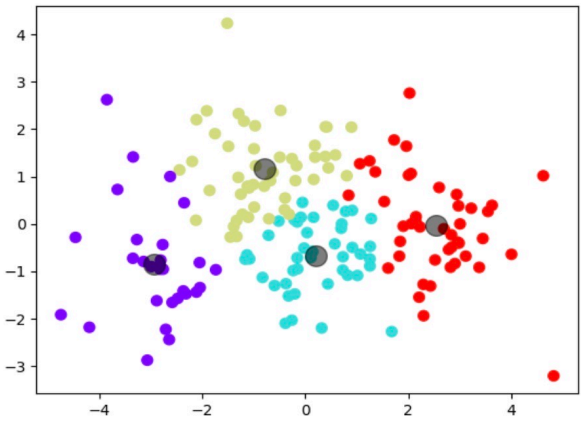
GeoCluster for HELP International

IMPLEMENTAZIONE PT.4

Da cui si è deciso di applicare il K-Means prima con k = 4 e poi con k = 5 per scegliere, infine, la prima opzione, come di seguito illustrato, i punti neri rappresentano i centroidi.

```
 #Trovato il numero di cluster ideali possiamo eseguire il k-means
 kmeans = KMeans(n_clusters=4)
 kmeans.fit(pca_df)
 y_kmeans = kmeans.predict(pca_df)
 labels = kmeans.labels_

#mostriamo come sono stati formati i cluster
plt.scatter(pca_df.iloc[:, 0], pca_df.iloc[:, 1], c=y_kmeans, s=50, cmap='rainbow')
centers = kmeans.cluster_centers_
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
dataset = pca_df
#i pallini neri rappresentano i centroidi
print("Coefficiente di forma: %0.3f" % metrics.silhouette_score(pca_df, labels))
```



GeoCluster for HELP International

IMPLEMENTAZIONE PT.5

Come ultima parte d'implementazione è stato creato un dizionario contenente delle coppie (id - cluster_id).
Dove 'id' rappresenta l'identificativo della nazione e 'cluster_id' rappresenta l'identificativo del cluster dove è contenuta la Nazione.

```
#Creiamo un dizionario per memorizzare le associazioni istanza - cluster
kmeans_dic = []

#Recuperiamo e memorizziamo le associazioni
for c_row in range(len(dataset)):
    kmeans_dic.append({"id": c_row , "cluster_id": labels[c_row]})

#visualizziamo le coppie tupla - cluster
kmeans_dic
```

```
[{'id': 0, 'cluster_id': 3},
 {'id': 1, 'cluster_id': 1},
 {'id': 2, 'cluster_id': 1},
 {'id': 3, 'cluster_id': 3},
 {'id': 4, 'cluster_id': 2},
 {'id': 5, 'cluster_id': 1},
 {'id': 6, 'cluster_id': 1},
 {'id': 7, 'cluster_id': 0},
 {'id': 8, 'cluster_id': 0},
 {'id': 9, 'cluster_id': 1},
```

GeoCluster for HELP International

VALUTAZIONI FINALI

All' interno del Main abbiamo la possibilità di creare una Nazione con i suoi relativi dati e aggiungerla all'interno del Dataset, successivamente si chiama la funzione eseguiClustering() passando come parametro il Dataset aggiornato.
Questa funzione esegue nuovamente il K-MEANS ritornando una lista di Nazioni che fanno parte del cluster della nazione precedentemente aggiunta.
Successivamente si procede a stampare a video tutte le nazioni contenute nella lista ritornata dalla funzione.

```
Nazione Aggiunta:  [164, 'America', 20.0, 99.0, 9.12, 20, 1456, 1.22, 87.0, 1.7, 1600]
id_cluster dell elemento aggiunto:  0
[4, 10, 11, 13, 14, 16, 18, 20, 24, 27, 30, 41, 42, 43, 51, 52, 57, 65, 67, 78, 83, 85, 86, 89, 90,
128, 129, 131, 132, 133, 135, 140, 145, 149, 151, 153, 159, 161, 164]
Ecco le nazioni del cluster:
Antigua and Barbuda
Bahamas
Bahrain
Barbados
Belarus
Belize
```

Repository: github.com/ocone28/ClusteringCountriesFIA