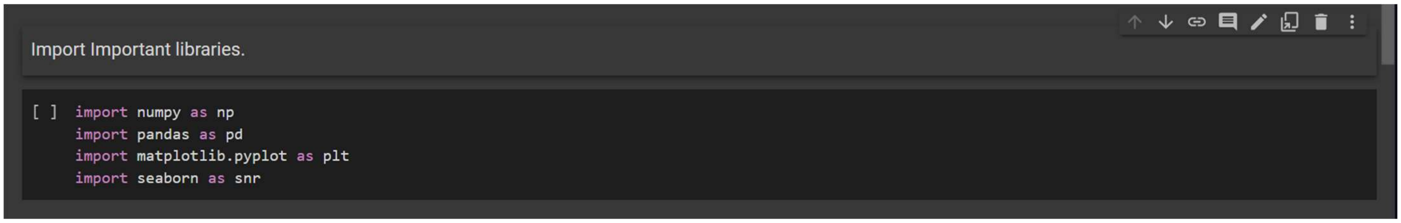
Shahjan
Khan

Project on Country Clustering using Machine Learning Models

- **Aim:** To create a Data science project where problem statement is like that, Help International have been able to raise around $ 10 million. Now the CEO of NGO needs to decide how to use this money strategically and effectively. So, CEO of the NGO have to make decision to choose the countries that are in the direst need of aid. Hence, we categorize the country the countries using some socio-economic and health factors that determine the overall development of the country. Then suggest the countries which the CEO needs to focus on the most.
- **Steps to be taken in the Project is sub-divided into the following sections. These are:**
  - Loading necessary libraries such as numpy , pandas , sklearn etc.
  - Loading and showing dataset as CSV file.
  - Take the necessary features from data.
  - Visualization pattern of the features with respect to countries.
  - Make Dendrogram for Clusters of Countries.
  - Import different kind of clustering algorithms and make clusters.
  - training those models using .fit() .
  - Predicting the trained models using .fit_predict().
  - Use the other model of clustering and do same process then check accuracy.
  - Finally There are clusters of countries for CEO to check where he should focus more.

- **Step-1:** Loading the necessary libraries such as numpy , pandas , sklearn etc.

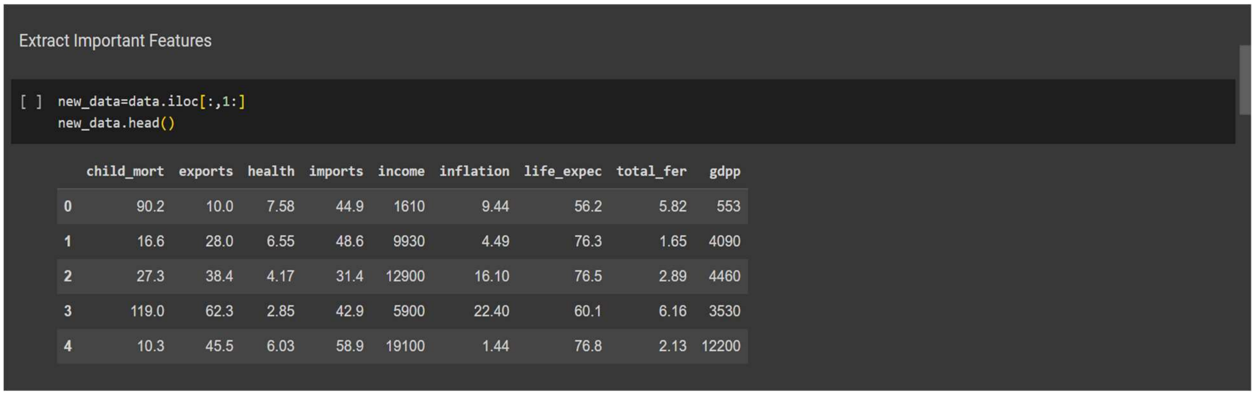


Import Important libraries.

```
[ ]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as snr
```

Shahjan
Khan

- **Step-2:** Loading and showing dataset as CSV file.

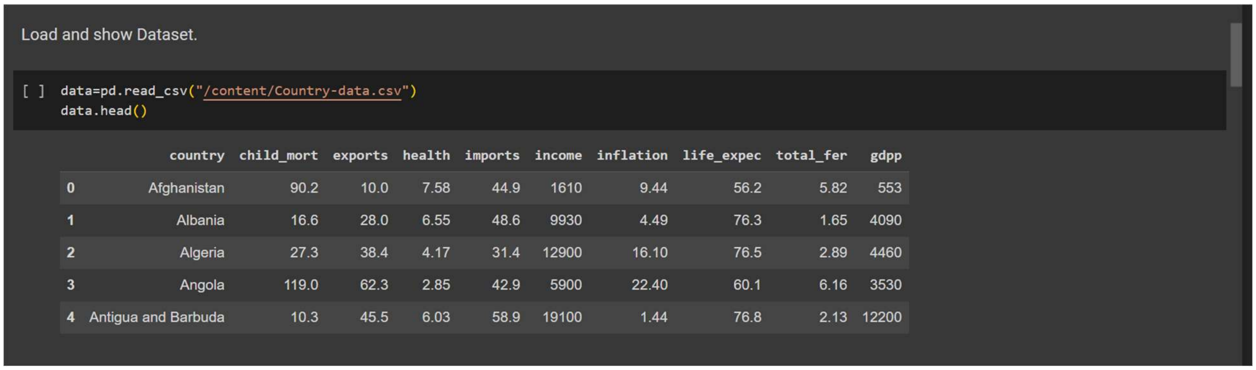


Extract Important Features

```
[ ] new_data=data.iloc[:,1:]
    new_data.head()
```

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

- **Step-3:** Take the necessary features from data.

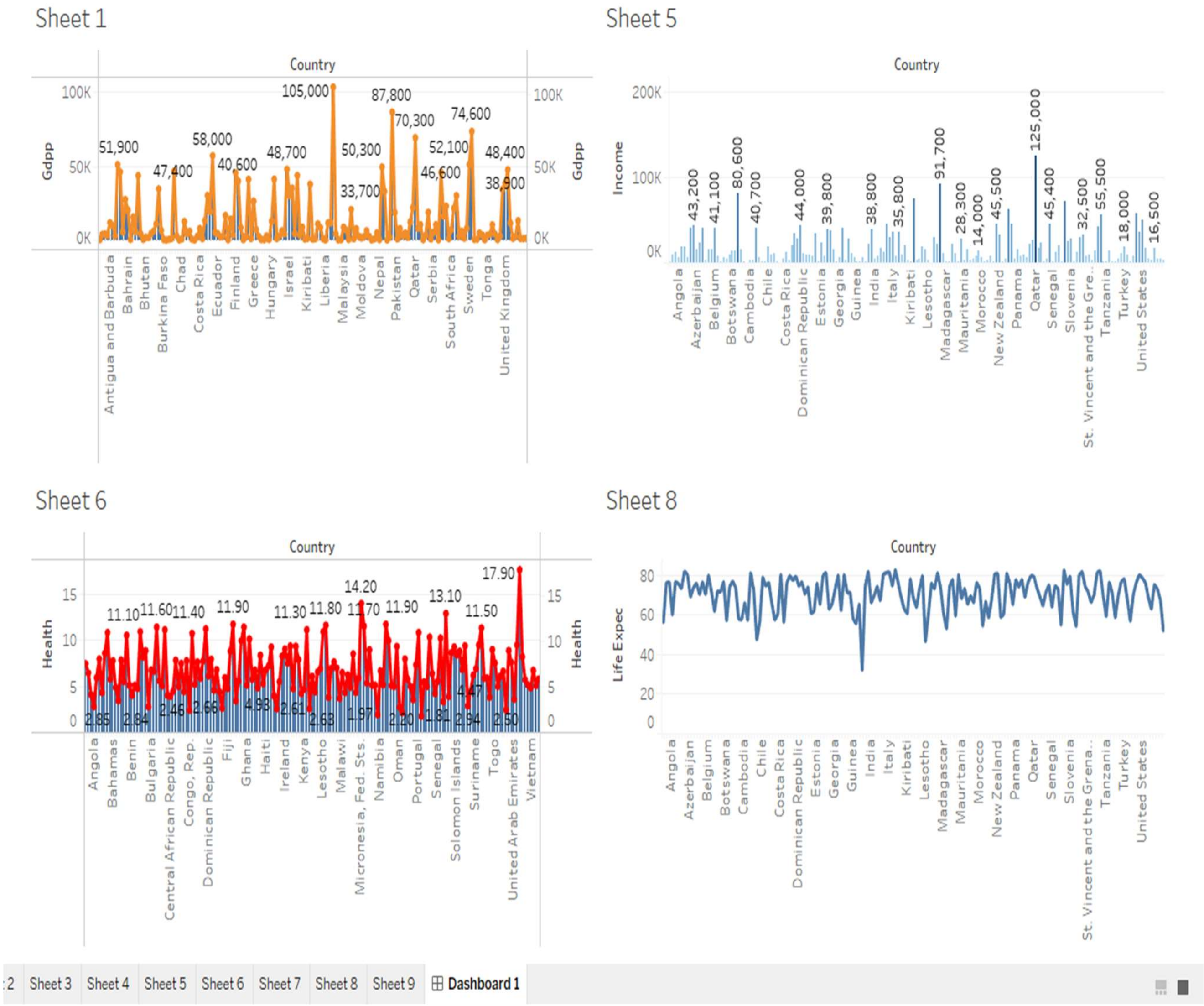


Load and show Dataset.

```
[ ] data=pd.read_csv("/content/Country-data.csv")
    data.head()
```

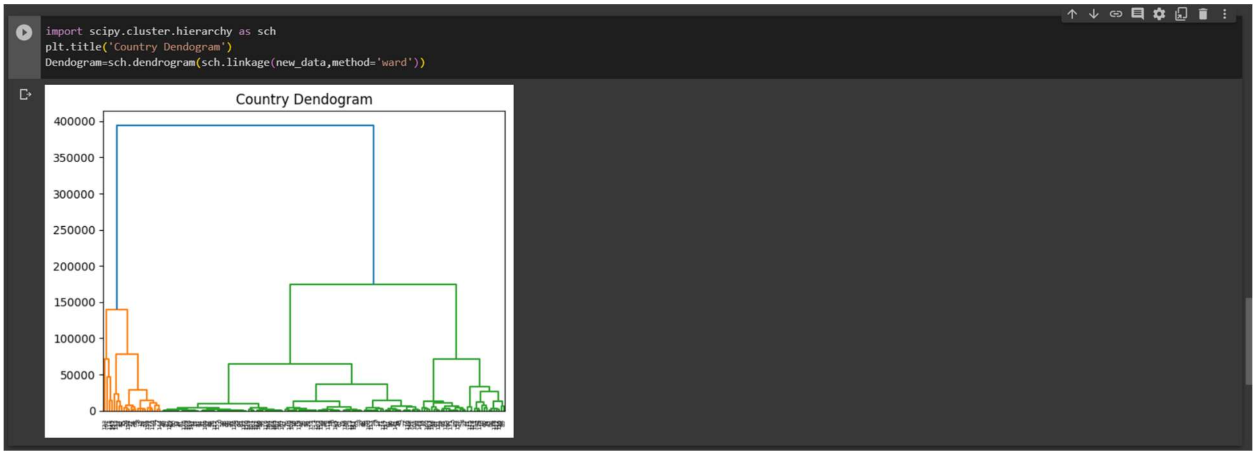
	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	10.0	7.58	44.9	1610	9.44	56.2	5.82	553
1	Albania	16.6	28.0	6.55	48.6	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	38.4	4.17	31.4	12900	16.10	76.5	2.89	4460
3	Angola	119.0	62.3	2.85	42.9	5900	22.40	60.1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

- **Step-4:** Visualization pattern of the features with respect to countries.



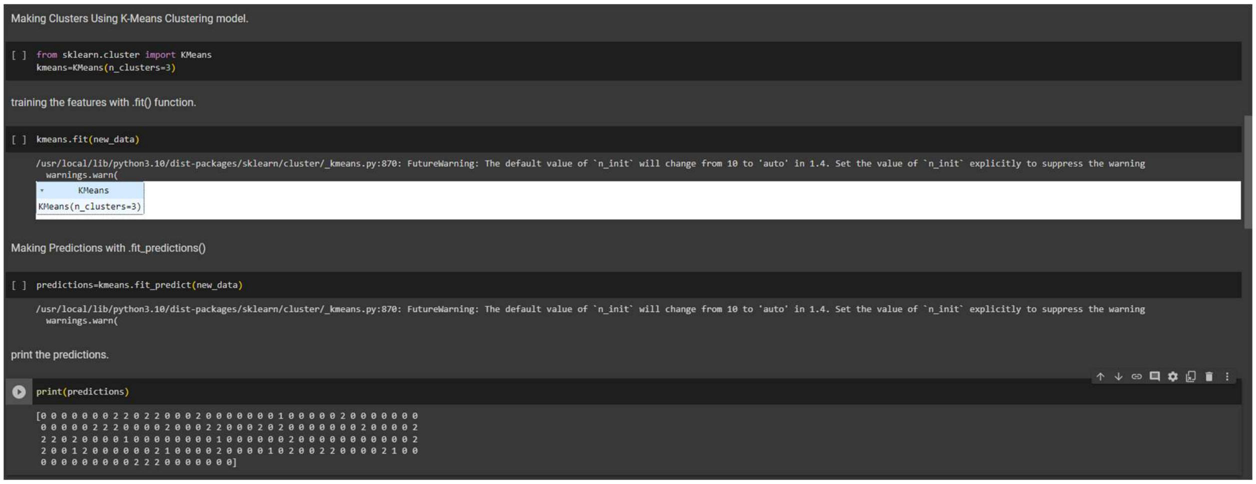
Shahjan
Khan

- **<u>Step-5</u>:** Make Dendrogram for Clusters of Countries.



```
import scipy.cluster.hierarchy as sch
plt.title('Country Dendogram')
Dendogram=sch.dendrogram(sch.linkage(new_data,method='ward'))
```

- **<u>Step-6</u>:** Import different kind of clustering algorithms and make clusters
    1. K-Means Clustering -



Making Clusters Using K-Means Clustering model.

```
[ ] from sklearn.cluster import KMeans
    kmeans=KMeans(n_clusters=3)
```

training the features with .fit() function.

```
[ ] kmeans.fit(new_data)

    /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
      warnings.warn(
    ▾       KMeans
    KMeans(n_clusters=3)
```

Making Predictions with .fit_predictions()

```
[ ] predictions=kmeans.fit_predict(new_data)

    /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress the warning
      warnings.warn(
```

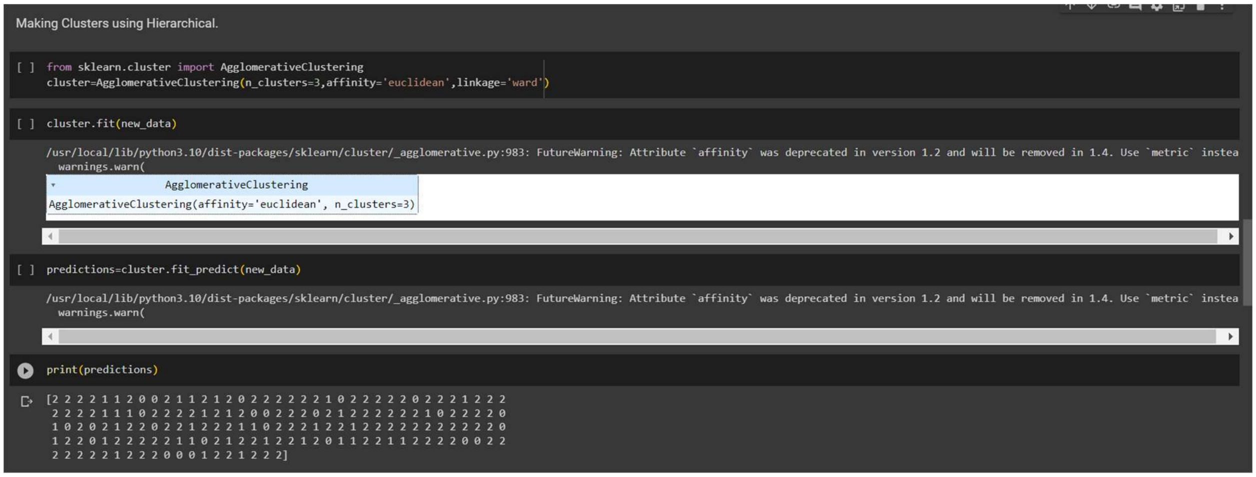
print the predictions.

```
print(predictions)

[0 0 0 0 0 0 0 2 2 0 2 2 0 0 0 2 0 0 0 0 0 0 0 1 0 0 0 0 0 2 0 0 0 0 0 0 0
 0 0 0 0 0 2 2 2 0 0 0 0 2 0 0 0 2 2 0 0 0 2 0 2 0 0 0 0 0 0 0 2 0 0 0 0 2
 2 2 0 2 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 2
 2 0 0 1 2 0 0 0 0 0 0 2 1 0 0 0 0 2 0 0 0 0 1 0 2 0 0 2 2 0 0 0 0 2 1 0 0
 0 0 0 0 0 0 0 0 0 2 2 2 0 0 0 0 0 0 0]
```

Shahjan
Khan

2. Hierarchical Clustering.



Making Clusters using Hierarchical.

```
[ ] from sklearn.cluster import AgglomerativeClustering
    cluster=AgglomerativeClustering(n_clusters=3,affinity='euclidean',linkage='ward')
```

```
[ ] cluster.fit(new_data)

    /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_agglomerative.py:983: FutureWarning: Attribute `affinity` was deprecated in version 1.2 and will be removed in 1.4. Use `metric` instea
      warnings.warn(
    ▾            AgglomerativeClustering
    AgglomerativeClustering(affinity='euclidean', n_clusters=3)
```

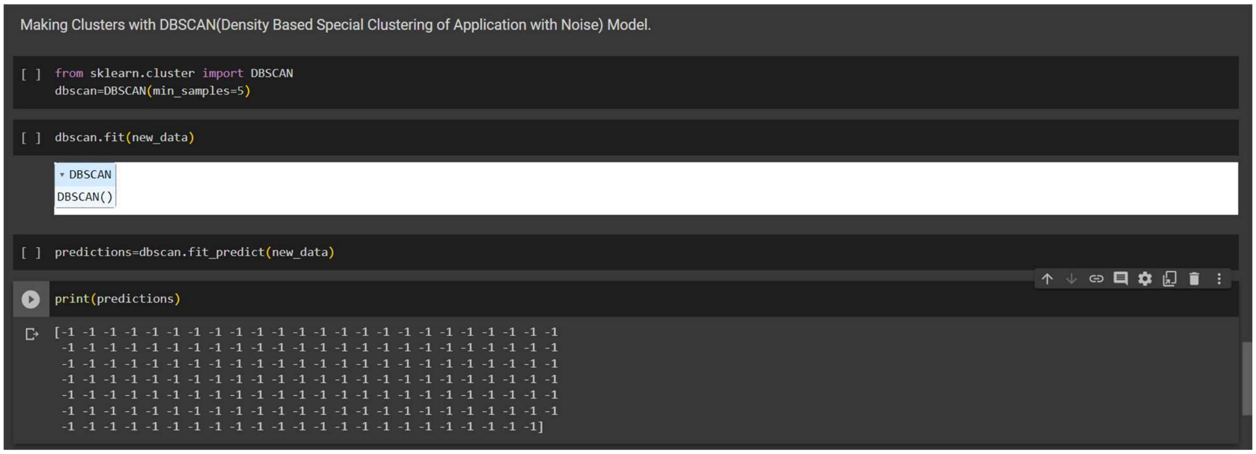
```
[ ] predictions=cluster.fit_predict(new_data)

    /usr/local/lib/python3.10/dist-packages/sklearn/cluster/_agglomerative.py:983: FutureWarning: Attribute `affinity` was deprecated in version 1.2 and will be removed in 1.4. Use `metric` instea
      warnings.warn(
```

```
print(predictions)

[2 2 2 2 1 1 2 0 0 2 1 1 2 1 2 0 2 2 2 2 2 2 1 0 2 2 2 2 2 0 2 2 2 1 2 2 2
 2 2 2 2 1 1 1 0 2 2 2 2 1 2 1 2 0 0 2 2 2 0 2 1 2 2 2 2 2 2 1 0 2 2 2 2 0
 1 0 2 0 2 1 2 2 0 2 2 1 2 2 2 1 1 0 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 2 0
 1 2 2 0 1 2 2 2 2 2 1 1 0 2 1 2 2 1 2 2 1 2 0 1 1 2 2 1 1 2 2 2 2 0 0 2 2
 2 2 2 2 2 1 2 2 2 0 0 0 1 2 2 1 2 2 2]
```

3. DBSCAN Clustering.



Making Clusters with DBSCAN(Density Based Special Clustering of Application with Noise) Model.

```
[ ] from sklearn.cluster import DBSCAN
    dbscan=DBSCAN(min_samples=5)
```

```
[ ] dbscan.fit(new_data)

    ▾ DBSCAN
    DBSCAN()
```

```
[ ] predictions=dbscan.fit_predict(new_data)
```

```
print(predictions)

[-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1]
```

<u>Conclusion:</u> - From this project we make Clusters of countries which is in very direst condition and which countries should the CEO more focus for help. By this project, we can suggest the CEO the right cluster(area) of countries for donation.