Name / Mostafa Mohamed Ismail Ahmed

ID / 20191613931

Group / 4A

The project is all on my own

Dataset Link :

https://www.kaggle.com/ajaypalsinghlo/world-happiness-report-2021?select=world-happiness-report-2021.csv

---

Dataset description :

The World Happiness Report is a landmark survey of the state of global happiness . The report continues to gain global recognition as governments, organizations and civil society increasingly use happiness indicators to inform their policy-making decisions. Leading experts across fields – economics, psychology, survey analysis, national statistics, health, public policy and more – describe how measurements of well-being can be used effectively to assess the progress of nations. The reports review the state of happiness in the world today and show how the new science of happiness explains personal and national variations in happiness.

The happiness scores and rankings use data from the Gallup World Poll. The columns following the happiness score estimate the extent to which each of six factors – economic production, social support, life expectancy, freedom, absence of corruption, and generosity – contribute to making life evaluations higher in each country than they are in Dystopia, a hypothetical country that has values equal to the world's lowest national averages for each of the six factors. They have no impact on the total score reported for each country, but they do explain why some countries rank higher than others.

Target from this application :

We want to explain the relation between **Healthy life expectancy (Rank of the country based on the Happiness Score)** and **Generosity .**

---

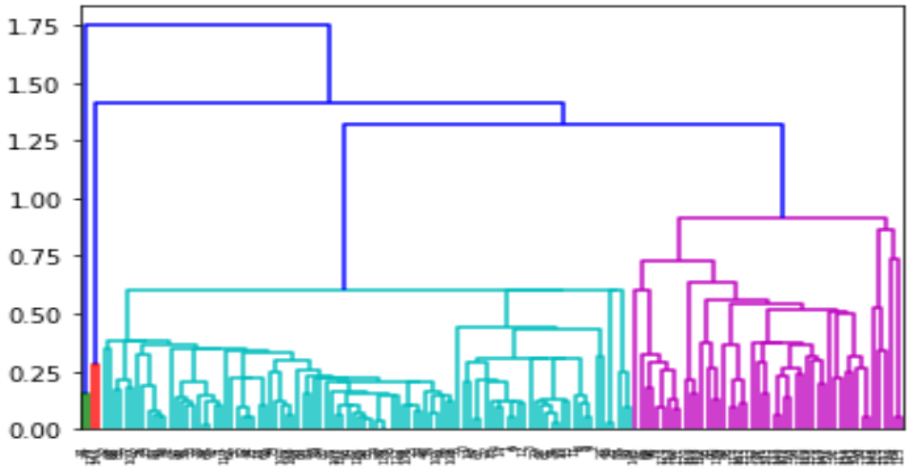
plotted graphs for each algorithm :

i used " matplotlib.pyplot as plt " library to scattering data

1)hierarchial clustering:

1) dendrogram

```
#shows the hierarchical relationship between objects.
dendrogram= sch.dendrogram(sch.linkage(data1,'single'))
```



2) scattering data

2) K-Medoids clustering:

1) scattering data

```
#scattering data
plt.scatter(data2[:,0], data2[:,1], c=cluster.labels_, cmap='rainbow')
plt.title('happiness_data')
plt.xlabel('Healthy life expectancy(48.478-76.953)')
plt.ylabel('Generosity(-0.288-0.542))')
```

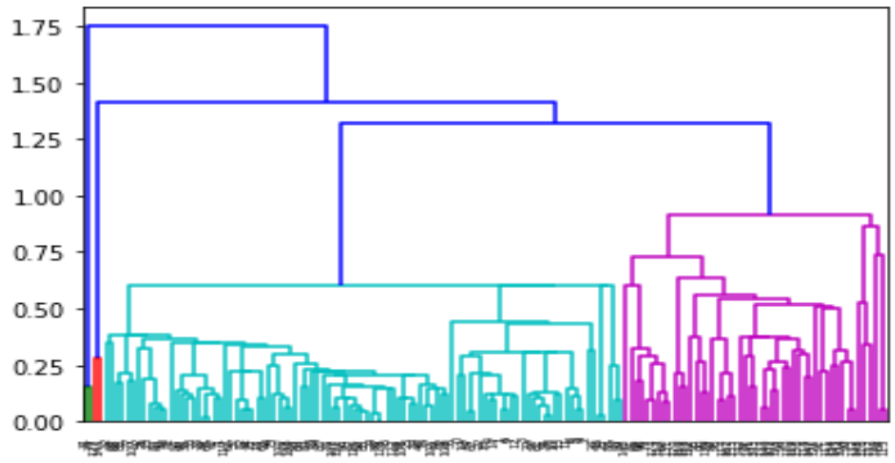
```
Text(0,0.5,'Generosity(-0.288-0.542))')
```

Explain results and insight by describing plotted graphs :

1)hierarchial clustering:

1)

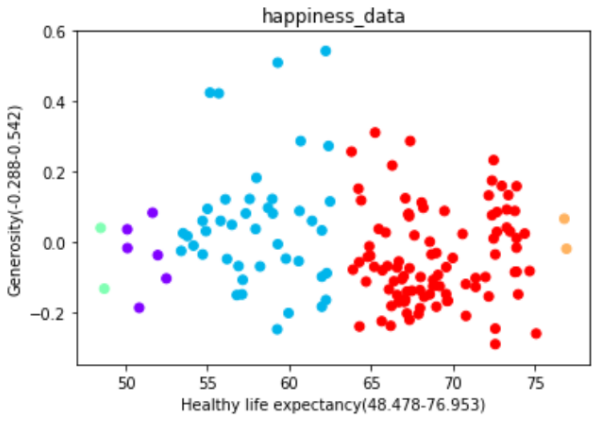
```
#shows the hierarchical relationship between objects.
dendrogram= sch.dendrogram(sch.linkage(data1,'single'))
```



2)

```
#scattering data
plt.scatter(data1[:,0], data1[:,1], c=cluster.labels_, cmap='rainbow')
plt.title('happiness_data')
plt.xlabel('Healthy life expectancy(48.478-76.953)')
plt.ylabel('Generosity(-0.288-0.542)')
```

```
Text(0,0.5,'Generosity(-0.288-0.542)')
```



In 1:

Used Agglomerative Hierarchical Clustering

1) The algorithm starts by finding the two points that are closest to each other on the basis of Euclidean distance.
   The vertical height of the dendogram shows the Euclidean distances between points.

2) The next step is to join the cluster formed by joining two points to the next nearest cluster or point which in turn results in another cluster.
   This process continues until all the points are joined together to form one big cluster.

3) Once one big cluster is formed, the longest vertical distance without any horizontal line passing through it is selected and a horizontal line is drawn through it. The number of vertical lines this newly created horizontal line passes is equal to number of clusters.

   Take a look at the plot 1 .

In 2:

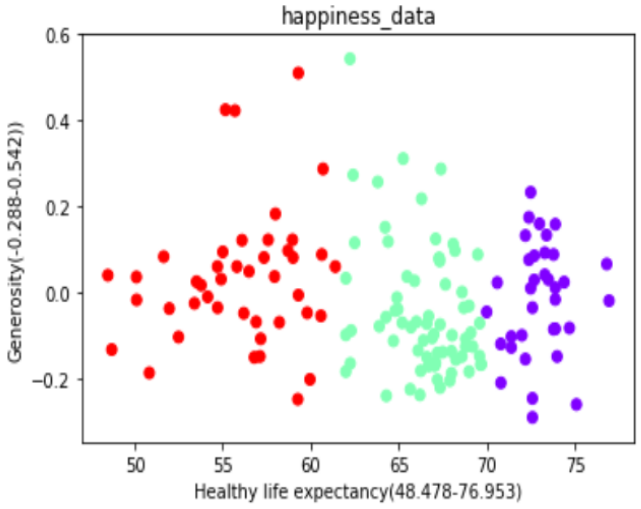
You can see the data points in the form of five clusters

1) Green data points refers Generosity which Rank of the country based on the Happiness Score is the smallest from **first to 50**
2) purple data points refers Generosity which Rank of the country based on the Happiness Score **from 50 to until before 55**
3) blue data points refers Generosity which Rank of the country based on the Happiness Score **From after 50 to before 65 (average) (The second largest cluster)**
4) Red data points refers Generosity which Rank of the country based on the Happiness Score **From before 65 (average) to 75 (The largest cluster)**
5) orange data points refers Generosity which Rank of the country based on the Happiness Score **From 75 to end (The Smallest cluster)**

2) K-Medoids clustering:

```
#scattering data
plt.scatter(data2[:,0], data2[:,1], c=cluster.labels_, cmap='rainbow')
plt.title('happiness_data')
plt.xlabel('Healthy life expectancy(48.478-76.953)')
plt.ylabel('Generosity(-0.288-0.542))')
```

```
Text(0,0.5,'Generosity(-0.288-0.542))')
```



Description of K-Medoids clustering:

1) The k-medoids algorithm is a clustering approach related to k-means clustering for partitioning a data set into k groups or clusters. In k-medoids clustering, each cluster is represented by one of the data point in the cluster. These points are named cluster medoids.

2) The term medoid refers to an object within a cluster for which average dissimilarity between it and all the other the members of the cluster is minimal. It corresponds to the most centrally located point in the cluster. These objects (one per cluster) can be considered as a representative example of the members of that cluster which may be useful in some situations. Recall that, in k-means clustering, the center of a given cluster is calculated as the mean value of all the data points in the cluster.
3) K-medoid is a robust alternative to k-means clustering. This means that, the algorithm is less sensitive to noise and outliers, compared to k-means, because it uses medoids as cluster centers instead of means (used in k-means).

K-medoids Algorithm :

1) For a given cluster assignment C ,find the observation in the cluster minimizing the total distance to other points in that cluster
2) Calculate the distance between each point and all other points.
3) Given a set of cluster centers , minimize the total error by assigning each observation to the closest (current) cluster center.
4) Repeat for each point
5) The medoids is the point which total distance is less than the others.

From Scatter plot :

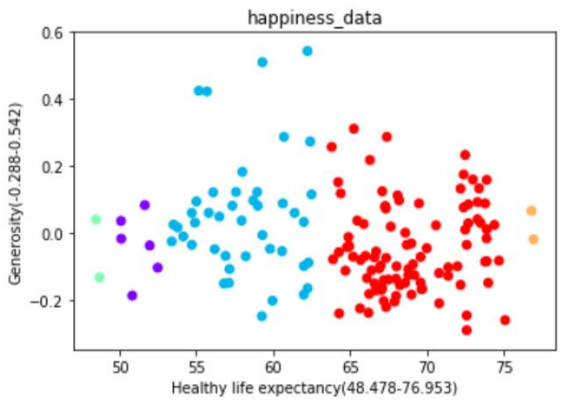
We classify data into 3 clusters :

1) Red data points refers Generosity From(-0.2 to before 0.6) which Rank of the country based on the Happiness Score from **first to average (The largest cluster) Contain the lower Rank of the country based on the Happiness Score**
2) Green data points refers Generosity From(-0.2 to before 0.6) which Rank of the country based on the Happiness Score **from after 60 to 70**
3) purple data points refers Generosity From(-0.2 to before 0.4) which Rank of the country based on the Happiness Score **From 70 to End Contain the largest Rank of the country based on the Happiness Score**

Comparison between Agglomerative Hierarchical Clustering Algorithm, and K-Medoids Algorithm results :

Agglomerative Hierarchical Clustering Algorithm	K-Medoids Algorithm
Used : Dendograms to define number of cluter = 5 , and scatter plot to visualize data **Advantages:** 1) No apriori information about the number of clusters required. 2) Easy to implement and gives best result in some cases . **Disadvantages:** 1) Algorithm can never undo what was done previously. 2) Time complexity of at least O($n^2 \log n$) is required, where *'n'* is the number of data points. 3) Based on the type of distance matrix chosen for merging different algorithms can suffer with one or more of the following: 1) Sensitivity to noise and outliers 2) Breaking large clusters 3) Difficulty handling different sized clusters and convex shapes 4) No objective function is directly minimized 5) Sometimes it is difficult to identify the correct number of clusters by the dendogram.	**Used :** cluter = 3 , and scatter plot to visualize data **Advantages:** 1. It is simple to understand and easy to implement. 2. K-Medoid Algorithm is fast and converges in a fixed number of steps. 3. PAM is less sensitive to outliers than other partitioning algorithms. **Disadvantages:** 1. The main disadvantage of K-Medoid algorithms is that it is not suitable for clustering non-spherical (arbitrary shaped) groups of objects. This is because it relies on minimizing the distances between the non-medoid objects and the medoid (the cluster centre) – briefly, it uses compactness as clustering criteria instead of connectivity. 2. It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.

Output :



Output :