PENGGUNAAN K-MEANS CLUSTERING
UNTUK MENGANALISIS PERKEMBANGAN NEGARA-NEGARA

Oleh:

Farhan Rangkuti (1304202025)

Kelas:

IF-45-01.1PJJ

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM

BANDUNG

2023

STATEMENT:

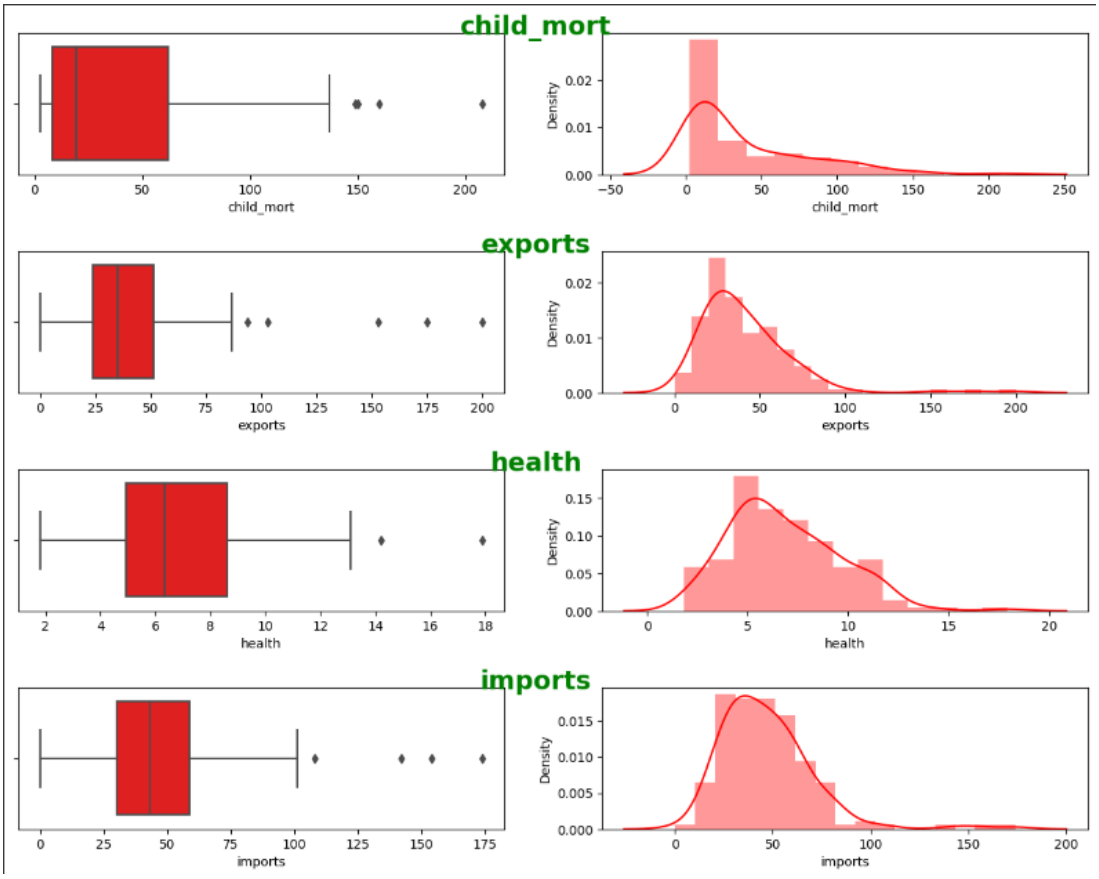
I do this assignment honestly and follow the rule of conduct of academic ethics in case I do not follow the rule I am ready to get a 0 (zero) mark for my assignment.

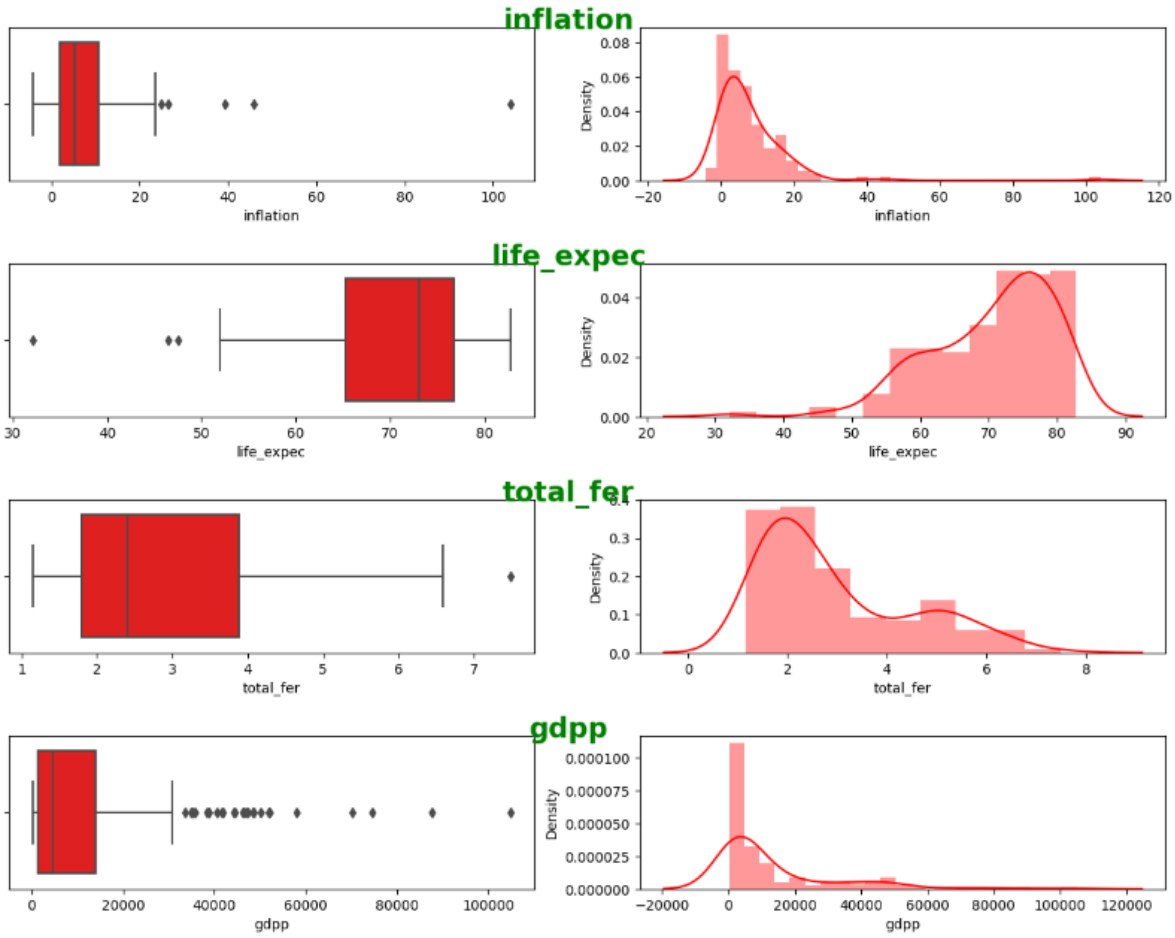
Part 1: Overview of the chosen data sets

Dataset yang digunakan adalah Country-data.csv yang bersumber dari Kaggle dengan link sebagai berikut:https://www.kaggle.com/datasets/rohan0301/unsupervised-learning-on-country-data

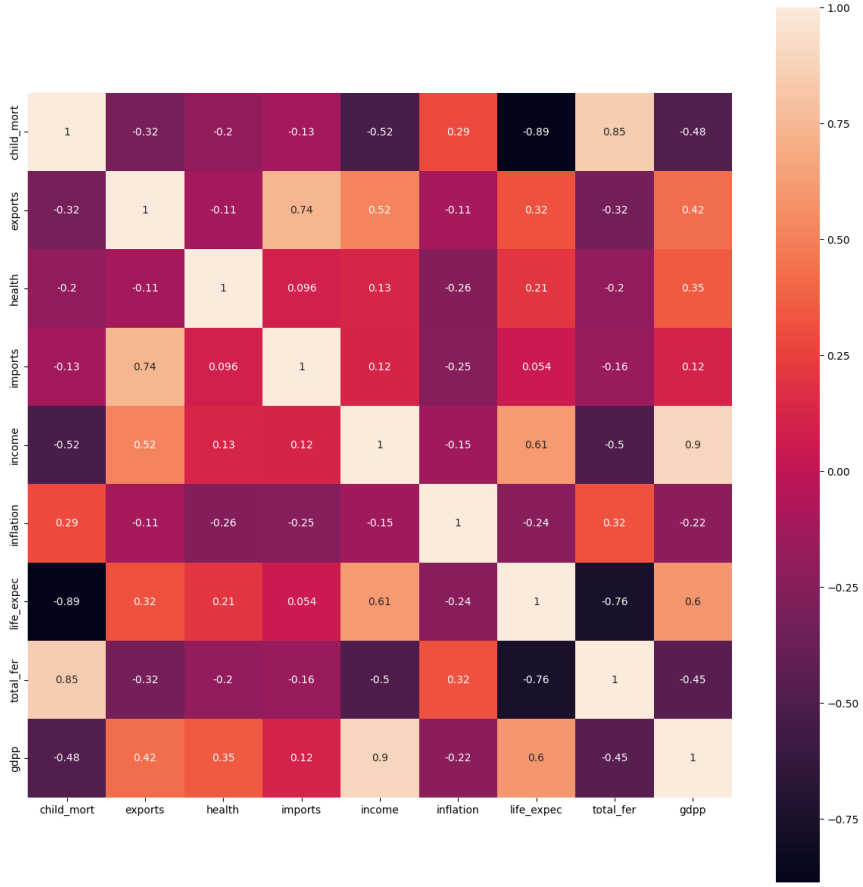
Data ini bertujuan untuk mengkategorikan negara-negara berdasarkan faktor sosio-ekonomis dan faktor kesehatannya untuk menentukan keseluruhan perkembangan negara tersebut. Data ini berisi 167 baris dan 10 kolom yang masing-masing memiliki penjelasan sebagai berikut:

country	Name of the country
child_mort	Death of children under 5 years of age per 1000 live births
exports	Exports of goods and services per capita. Given as %age of the GDP per capita
health	Total health spending per capita. Given as %age of GDP per capita
imports	Imports of goods and services per capita. Given as %age of the GDP per capita
Income	Net income per person
Inflation	The measurement of the annual growth rate of the Total GDP
life_expec	The average number of years a new born child would live if the current mortality patterns are to remain the same
total_fer	The number of children that would be born to each woman if the current age-fertility rates remain the same.
gdpp	The GDP per capita. Calculated as the Total GDP divided by the total population.





Data tidak memiliki missing values. Terdapat beberapa outlier pada data tetapi karena terdapat kemungkinan bahwa data tersebut memiliki informasi yang penting untuk pengklasifikasian data maka outlier tersebut tidak dihilangkan. Untuk distribusi data sendiri dapat dilihat bahwa hampir semua kolom memiliki distribusi positif atau condong ke kanan, sedangkan untuk kolom life_expec memiliki distribusi negatif atau condong ke kiri.



Dari heatmap di atas dapat dilihat bahwa beberapa kolom memiliki korelasi yang kuat, seperti :

child_mort & total_fer : 0.85

income & gdpp : 0.90

life_expec & child_mort : -0.89

untuk life_expec & child_mort sendiri memiliki korelasi negatif mengingat life expectancy dan child mortality merupakan dua kategori yang berkebalikan.

Berdasarkan investigasi saya terhadap kualitas dataset, saya akan melakukan preprocessing sebagai berikut:

- Data Cleaning : Tidak diperlukan karena tidak ada missing values atau outlier.
- Data Transformation : Melakukan normalisasi terhadap data menggunakan MinMaxScaler.
- Feature engineering : Menerapkan Principal Component analysis terhadap data untuk mengurangi jumlah fitur tetapi tetap menyimpan informasi yang penting dari data.

Part 2: Summary of the proposed data pre-processing

1. Data Transformation

   Pada tahap ini saya akan melakukan normalisasi terhadap data menggunakan metode MinMaxScaler pada library Scikit-Learn. Berikut merupakan alasan saya melakukan normalisasi terhadap data:

   - Fitur-fitur yang ada memiliki skala pengukuran yang berbeda-beda.
   - Range value dari fitur sangat bervariasi, ada yang sangat besar dan sangat kecil.
   - Dengan melakukan normalisasi kita dapat menghilangkan kemungkinan adanya bias terhadap dataset.

Normalization

```
[17] df = data.drop(columns='country', inplace=False)

     scaler = MinMaxScaler().fit_transform(df)
     df_norm = pd.DataFrame(scaler, columns=df.columns)
```

```
[18] df_norm
```

	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	0.426485	0.049482	0.358608	0.257765	0.008047	0.126144	0.475345	0.736593	0.003073
1	0.068160	0.139531	0.294593	0.279037	0.074933	0.080399	0.871795	0.078864	0.036833
2	0.120253	0.191559	0.146675	0.180149	0.098809	0.187691	0.875740	0.274448	0.040365
3	0.566699	0.311125	0.064636	0.246266	0.042535	0.245911	0.552268	0.790221	0.031488
4	0.037488	0.227079	0.262275	0.338255	0.148652	0.052213	0.881657	0.154574	0.114242
...	...	...	...	...	...	...	...	...	...
162	0.129503	0.232582	0.213797	0.302609	0.018820	0.063118	0.609467	0.370662	0.026143
163	0.070594	0.142032	0.192666	0.100809	0.127750	0.463081	0.854043	0.208202	0.126650
164	0.100779	0.359651	0.312617	0.460715	0.031200	0.150725	0.808679	0.126183	0.010299
165	0.261441	0.149536	0.209447	0.197397	0.031120	0.257000	0.698225	0.555205	0.010299
166	0.391918	0.184556	0.253574	0.177275	0.021473	0.168284	0.392505	0.670347	0.011731

167 rows × 9 columns

Dapat dilihat di atas kita melakukan normalisasi pada dataset ‘df’ yang berisi setiap kolom kecuali ‘country’ karena berbentuk string, kemudian hasilnya disimpan pada variabel df_norm yang hasilnya bisa kita lihat dibawah.

2. Feature Engineering.

   Pada tahap ini saya akan menggunakan teknik Principal Component Analysis (PCA) pada library Scikit-Learn untuk mengurangi dimensionalitas terhadap data tetapi tetap menyimpan sebagian besar informasi.

   - Mengurangi dimensionalitas data.
   - Tetap menyimpan sebagian besar informasi yang ada.
   - Mengurangi noise.

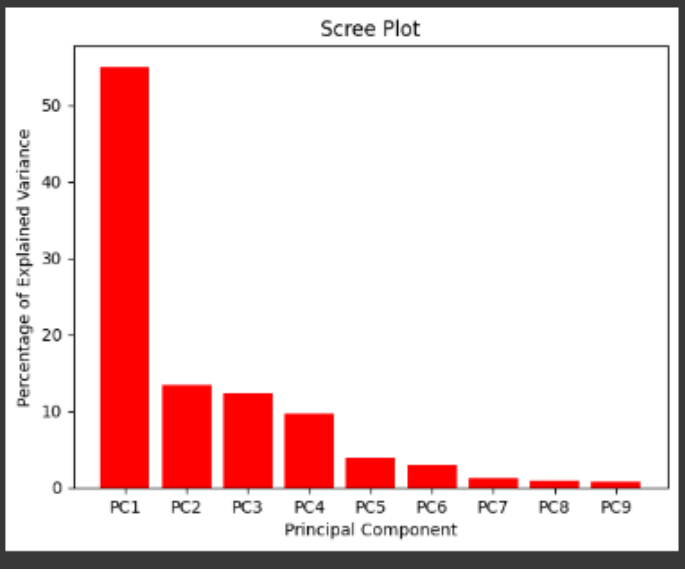
PCA

```
[57] pca = PCA(n_components=9).fit(df_norm)
     variance = pca.explained_variance_ratio_
     print(variance)
```

```
[0.55001227 0.13384784 0.12301053 0.09749047 0.03777964 0.03013659
 0.01190434 0.00887791 0.00694042]
```

```
[24]
     per_var = np.round(pca.explained_variance_ratio_*100, decimals =1)
     labels = ['PC' + str(x) for x in range (1, len(per_var)+1)]


     plt.bar(x=range(1,len(per_var)+1), height=per_var, tick_label = labels, color = 'red')
     plt.ylabel('Percentage of Explained Variance')
     plt.xlabel('Principal Component')
     plt.title('Scree Plot')
     plt.show()
```



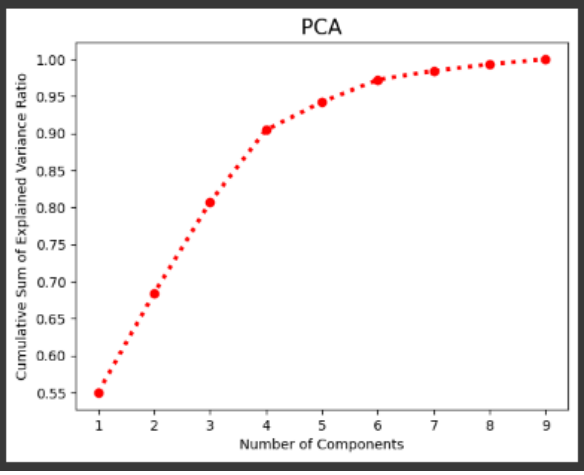
```
components = np.arange(1, len(variance) + 1)

plt.plot(components, np.cumsum(variance), linewidth=3, marker='o', linestyle='dotted', color='red')

plt.title("PCA", fontsize=15)
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Sum of Explained Variance Ratio')

plt.yticks(np.arange(0.55, 1.05, 0.05))

plt.show()
```



Dapat kita lihat pada plot di atas maupun di bawah bahwa terdapat 5 komponen yang dapat menjelaskan 94% dari data, oleh karena itu kita akan menggunakan 5 kolom tersebut saja untuk merepresentasikan seluruh data yang disimpan pada variabel df_pcs.

```
[27] increment_pca = IncrementalPCA(n_components=5).fit_transform(df_norm)
     pcs = np.transpose(increment_pca)

[28] df_pcs = pd.DataFrame({'PC1':pcs[0], 'PC2':pcs[1], 'PC3':pcs[2], 'PC4':pcs[3],'PC5':pcs[4],})

[29] df_pcs
```

	PC1	PC2	PC3	PC4	PC5
0	0.599065	0.095506	0.157437	0.024382	0.042560
1	-0.158455	-0.212345	-0.063709	0.061067	-0.017134
2	-0.003661	-0.135842	-0.133938	-0.133376	0.092225
3	0.650272	0.276016	-0.143024	-0.155593	0.083921
4	-0.200684	-0.065162	-0.100418	0.037680	0.030574
...	...	...	...	...	...
162	0.160056	-0.030100	-0.121936	0.065807	0.007747
163	-0.061196	-0.169616	-0.058942	-0.245296	0.112279
164	-0.115510	-0.032083	-0.195224	0.232604	0.040592
165	0.332967	-0.019417	-0.030055	-0.104661	0.145588
166	0.573833	0.109080	0.031989	-0.044807	-0.017517

167 rows × 5 columns

```
[30] df_pcs.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 167 entries, 0 to 166
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   PC1     167 non-null    float64
 1   PC2     167 non-null    float64
 2   PC3     167 non-null    float64
 3   PC4     167 non-null    float64
 4   PC5     167 non-null    float64
dtypes: float64(5)
memory usage: 6.6 KB
```

Part 3: Implementing the proposed algorithm

Untuk permasalahan ini saya menggunakan algoritma K-Means dikarenakan impementasinya simpel dan efisien. Disini saya tidak menggunakan library eksternal seperti Scikit-Learn.

```
data_kmeans = df_pcs.values.tolist()

k = 3
centroids = random.sample(data_kmeans, k)
```

Merubah data df_pcs sebelumnya menjadi list agar dapat diproses dan disimpan pada variabel data_kmeans, kemudian menginisialisasi nilai k menjadi 3, kemudian menginisialisasi centroid secara random pada data.

```
def euclidean_distance(a, b):
    return sum((x - y) ** 2 for x, y in zip(a, b)) ** 0.5
```

Merupakan fungsi untuk menghitung Euclidean distance.

```
while True:

    labels = []
    clusters = [[] for _ in range(k)]
    for point in data_kmeans:
        distances = [euclidean_distance(point, centroid) for centroid in centroids]
        closest_centroid_index = distances.index(min(distances))
        labels.append(closest_centroid_index)
        clusters[closest_centroid_index].append(point)


    new_centroids = [list(np.mean(cluster, axis=0)) for cluster in clusters]
```

Melakukan looping, kemudian menginisialisasi variabel labels yang berfungsi menyimpan label kluster pada setiap data point, kemudian melakukan inisialisasi variabel clusters yang berisi sebanyak ‘k’ list yang mana setiap list akan merepresentasikan setiap kluster.

Kemudian kita akan melakukan looping terhadap setiap data point yang ada pada data, kemudian kita akan menghitung Euclidean distance data point tersebut terhadap setiap centroid, kemudian hasilnya akan disimpan pada variabel distances yang berbentuk list. Kemudian melakukan penghitungan index centroid yang paling dekat yang akan menjadi label klaster pada data point saat ini. Kemudian index tersebut akan ditambahkan pada variabel labes untuk data point saat ini. Kemudian kita akan Menambahkan titik data saat ini ke dalam daftar titik untuk klaster yang diidentifikasi oleh closest_centroid_index.

```
new_centroids = [list(np.mean(cluster, axis=0)) for cluster in clusters]
```

Kemudian kita akan menghitung pusat centroid baru untuk setiap klaster berdasarkan rata-rata dari data point dalam klaster tersebut.

```
if new_centroids == centroids:
    break
else:
    centroids = new_centroids
```

Kemudian kita akan melakukan pengecekan terhadap konvergensi dengan membandingkan centroid baru ini dengan centroid sebelumnya apabila centroid tidak berubah maka program akan melakukan break pada loop dikarenakan terjadi konvergensi.

```
labels_df = pd.DataFrame({'Cluster': labels})

df_with_labels = pd.concat([df_pcs, labels_df], axis=1)

print("Final centroids:", centroids)
print("DataFrame with assigned labels:")
print(df_with_labels)
```

Program akan membuat dataframe baru yang berisi label untuk setiap baris data yang dinamakan Cluster, kemudian akan dilakukan concat untuk menambahkannya pada datasebelumnya dan disimpan pada variabel df_with_labels. Kemudian program akan melakukan print centroids dan df_with_labels. Berikut merupakan output dari program.

```
Final centroids: [[-0.47324866641333846, 0.14480852111866166, 0.11965444122560707, -0.0560943843427933, -0.009161150316732274], [-0.07613640616721414,
DataFrame with assigned labels:
          PC1       PC2       PC3       PC4       PC5  Cluster
0    0.599065  0.095506  0.157437  0.024382  0.042560        2
1   -0.158455 -0.212345 -0.063709  0.061067 -0.017134        1
2   -0.003661 -0.135842 -0.133938 -0.133376  0.092225        1
3    0.650272  0.276016 -0.143024 -0.155593  0.083921        2
4   -0.200684 -0.065162 -0.100418  0.037680  0.030574        1
..        ...       ...       ...       ...       ...      ...
162  0.160056 -0.030100 -0.121936  0.065807  0.007747        1
163 -0.061196 -0.169616 -0.058942 -0.245296  0.112279        1
164 -0.115510 -0.032083 -0.195224  0.232604  0.040592        1
165  0.332967 -0.019417 -0.030055 -0.104661  0.145588        2
166  0.573833  0.109080  0.031989 -0.044807 -0.017517        2

[167 rows x 6 columns]
```

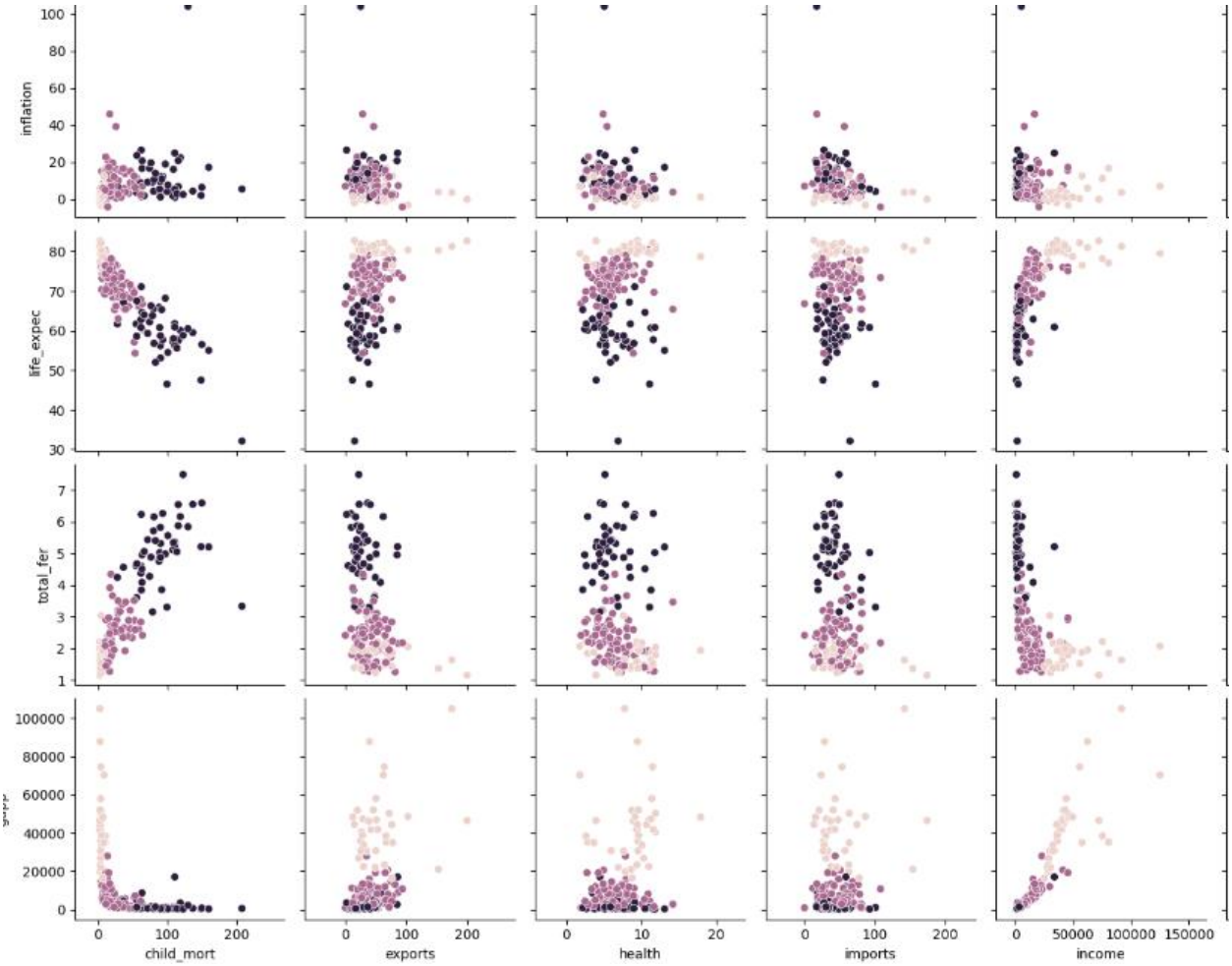
Google Colab Link :

https://colab.research.google.com/drive/1xlKok1i5fwRG-3uBb0iMM0eSllZX4ngz?usp=sharing

Reference :

Google Developers. (n.d.). Running the Algorithm. In Machine Learning Crash Course. Retrieved from [Google Developers]

Part 4: Evaluation of results



Cluster	child_mort	exports	gdpp	health	imports	income	inflation	life_expec	total_fer
0	4.897143	58.431429	43117.142857	8.917429	51.508571	45802.857143	2.535000	80.245714	1.741143
1	22.425581	40.382430	6719.790698	6.215581	46.932162	12770.813953	7.609023	72.582558	2.293256
2	93.284783	29.287174	1695.913043	6.338478	43.297826	3516.804348	12.097065	59.393478	5.090217

Gambar di atas merupakan plot distribusi kluster dan nilai rata-rata dari setiap kolom feature yang ada pada data.

	Country	PC1	PC2	PC3	PC4	PC5	Cluster	Status
0	Afghanistan	0.599065	0.095506	0.157437	0.024382	0.042560	2	Need Help
1	Albania	-0.158455	-0.212345	-0.063709	0.061067	-0.017134	1	Stable
2	Algeria	-0.003661	-0.135842	-0.133938	-0.133376	0.092225	1	Stable
3	Angola	0.650272	0.276016	-0.143024	-0.155593	0.083921	2	Need Help
4	Antigua and Barbuda	-0.200684	-0.065162	-0.100418	0.037680	0.030574	1	Stable
...	...	...	...	...	...	...	...	...
162	Vanuatu	0.160056	-0.030100	-0.121936	0.065807	0.007747	1	Stable
163	Venezuela	-0.061196	-0.169616	-0.058942	-0.245296	0.112279	1	Stable
164	Vietnam	-0.115510	-0.032083	-0.195224	0.232604	0.040592	1	Stable
165	Yemen	0.332967	-0.019417	-0.030055	-0.104661	0.145588	2	Need Help
166	Zambia	0.573833	0.109080	0.031989	-0.044807	-0.017517	2	Need Help

167 rows × 8 columns

Berdasarkan analisis hasil algoritma K-Means, data dapat diklasifikasikan menjadi ‘Do not need help’, ‘Stable’, ‘Need Help’. Alasan pengklasifikasian ini adalah sebagai berikut:

- Cluster 0 : Berisi nilai rata-rata positif untuk perkembangan ekonominya, ekpektansi hidup tinggi, dan angka kematian anak yang rendah. Kebanyakan merupakan negara dari Amerika Utara, Eropa, dan Australia.
- Cluster 1 : Berisi nilai rata-rata untuk perkembangan ekonomi, ekpektansi hidup, dan angka kematian anak. Kebanyakan merupakan negara dari Asia, Eropa, dan Amerika Selatan.
- Cluster 2 : Berisi nilai rata-rata Negatif untuk perkembangan ekonominya, ekpektansi hidup rendah, dan angka kematian anak yang tinggi. Kebanyakan merupakan negara dari Afrika.

Dapat disimpulkan bahwa Cluster 2 lebih membutuhkan bantuan dari NGO daripada Cluster lainnya berdasarkan faktor-faktor yang telah diperhitungkan sebelumnya.

Part 5: video presentation

Link Video :

https://drive.google.com/file/d/1xv7KVKHejD6Fk1ufMXq5ibk88W3Xs7sn/view?usp=sharing

Link Slide :

https://docs.google.com/presentation/d/1l4LcQp-yq5mJT2L_CX_nyd-vJvEt-xNQ/edit?usp=sharing&ouid=112445352741343745093&rtpof=true&sd=true

Link Laporan :

https://docs.google.com/document/d/10x-ulHrgF-et9HQoLs-8JCzNVGCXsMJ9/edit?usp=sharing&ouid=112445352741343745093&rtpof=true&sd=true