MACHINE LEARNING BANG ALGORITHMS

- NHÓM 5

Bố Cục

Tổng quan đề tài

Cơ sở lí thuyết

Tổng quan & Tiền xử lí dữ liệu

Áp dụng giải thuật

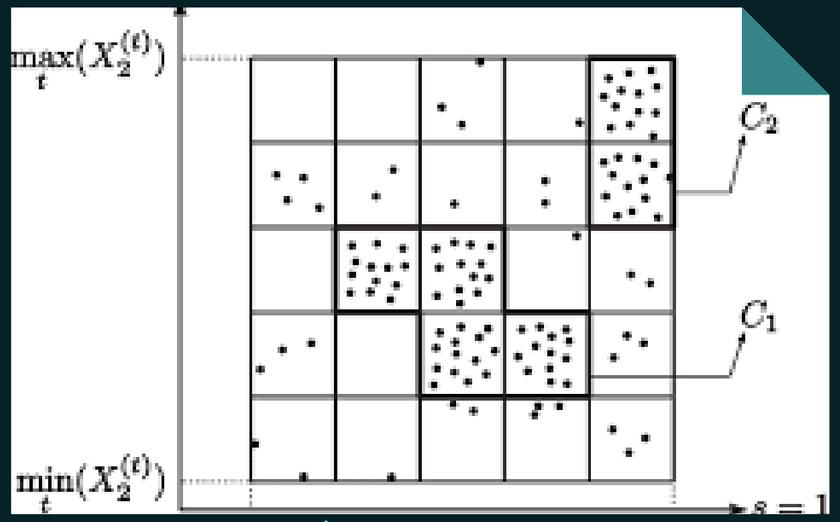
Kết luận & Đánh giá

TỔNG QUAN ĐỀ TÀI

Sơ lược về đề tài

Mục tiêu & Phương pháp nghiên cứu

CƠ SỞ LÍ THUYẾT



Mô Tả Tổng Quan

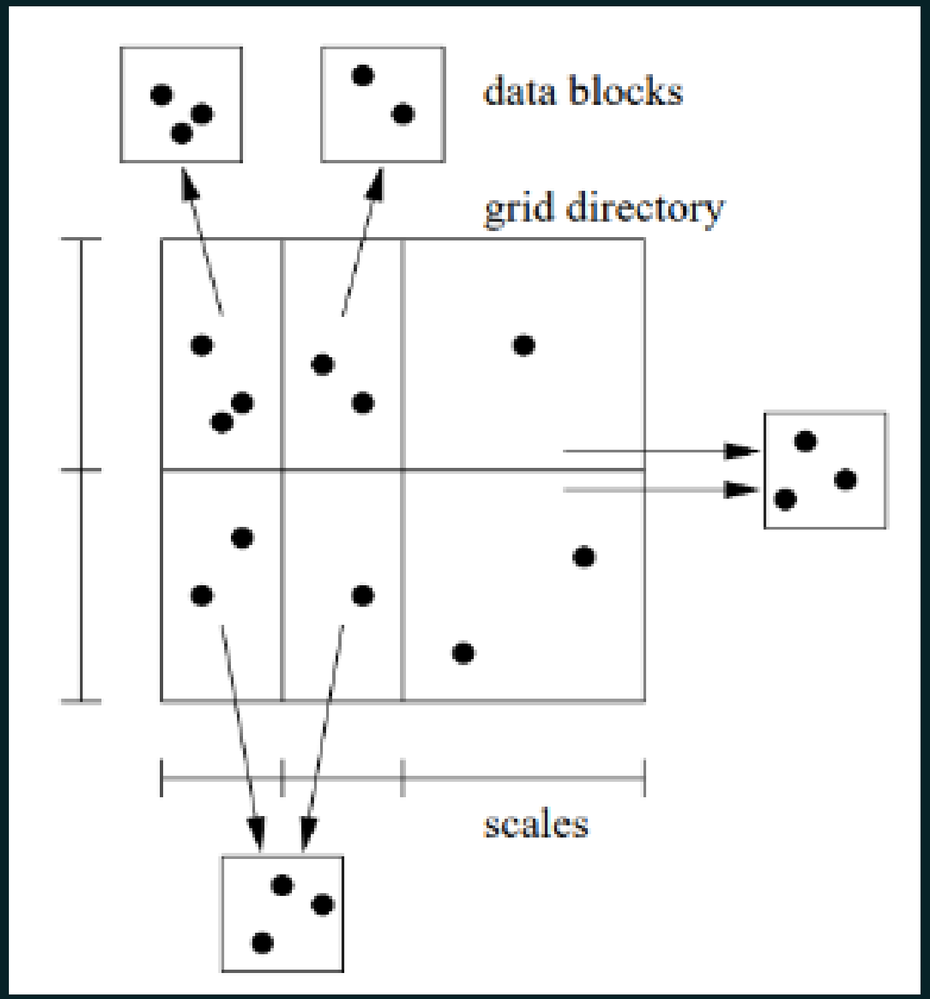
BANG được sử dụng để xử lý tập dữ liệu lớn để so sánh về mật độ dựa trên các mạng lưới ô nhằm tìm trung tâm và kết hợp các lưới ô xung quanh giúp gia tăng hiệu suất

CẤU TRÚC

Ý Tưởng

- BANG-Structure chia không gian giá trị thành các phần nhỏ hơn và quản lý các điểm dữ liệu bằng một tập hợp các khối hình chữ nhật xung quanh chúng
- Kết quả của quá trình gom cụm là việc tạo ra một Dendrogram như biểu đồ bên trái

BANG-Structure



Cấu trúc BANG phân chia không gian giá trị thành các khối hình chữ nhật và quản lý các điểm dữ liệu thông qua tập hợp giúp tạo ra một cấu trúc có tổ chức để hiệu quả phân chia mẫu bao gồm các thành phần sau:

- Thư mục lưới
- Vùng lưới
- Khối dữ liệu
- Vùng khối

Density Index Algorithm

Thuật toán này được sử dụng trong BANG-Clustering dùng để tính toán chỉ số mật độ cho từng khối dữ liệu trong không gian giá trị

$$V_B = \prod e_{B_i}, i = 1,2,3, \dots, k$$

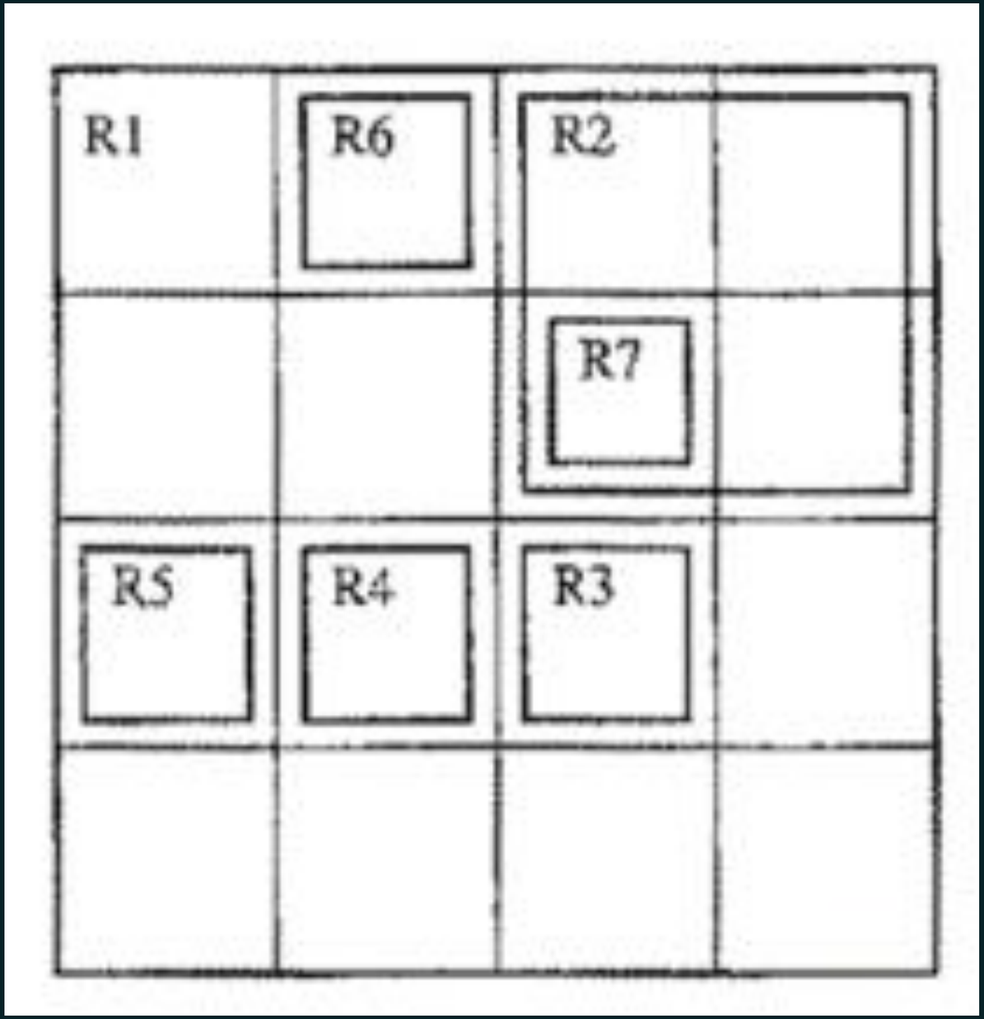
Sau khi tính xong thể tích của khối, nhóm tiến hành tính chỉ số mật độ của khối

$$D_B = \frac{P_B}{V_B}$$

Erich Schikuta and Martin Erhart, The BANG-Clustering System: Grid-Based Data Analysis

Sắp xếp các khối theo chỉ số mật độ giảm dần.Từ đó xây dựng các trung tâm cụm mới hoặc nhóm với các cụm hiện có

Neighbors



Trong cấu trúc BANG, có hai loại “neighborhood” có thể được phân biệt là “normal neighborhood” và “refined neighborhood”.

Dendrogram

Dendrogram được tính toán trực tiếp bởi thuật toán phân cụm BANG.

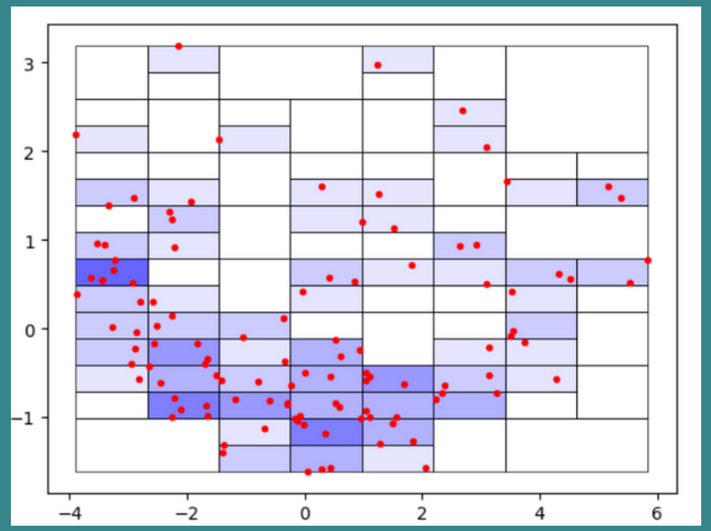
Nếu R1 là lân cận của R2 và R2 là lân cận của R3

- R1 > R2 > R3, thì xây dựng với R1, R2, và R3 một cụm (tìm kiếm lân cận bắt đầu từ R3).
- R1 > R2 < R3, thì xây dựng với R1, R2, và R3 một cụm (tìm kiếm lân cận bắt đầu từ R2).

Phương Pháp Thực Hiện

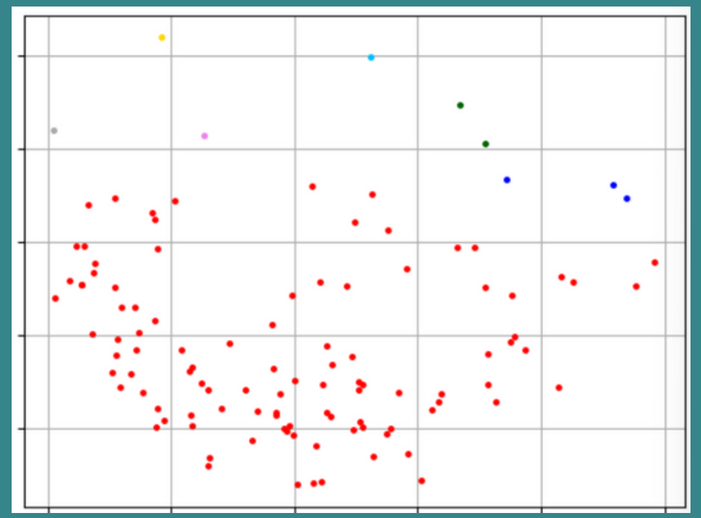
Bước 1

Khởi tạo cấu trúc lưới



Bước 2

Tính mật độ các khối dữ liệu



Bước 3

Sắp xếp khối dữ liệu theo chỉ số mật độ

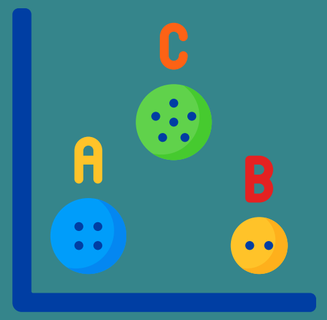
Phương Pháp Thực Hiện

Bước 4

Xác định khối trung tâm

Bước 5

Gộp khối dữ liệu



Bước 6

Lặp lại quá trình

Tổng quan bộ dữ liệu

‘Country Socioeconomic Data’ được lấy từ Kaggle, bộ dữ liệu cung cấp 1 loạt thông tin về hình thái kinh tế xã hội. Bộ dữ liệu bao gồm 10 thuộc tính với 167 quan sát, tương đương với 167 quốc gia

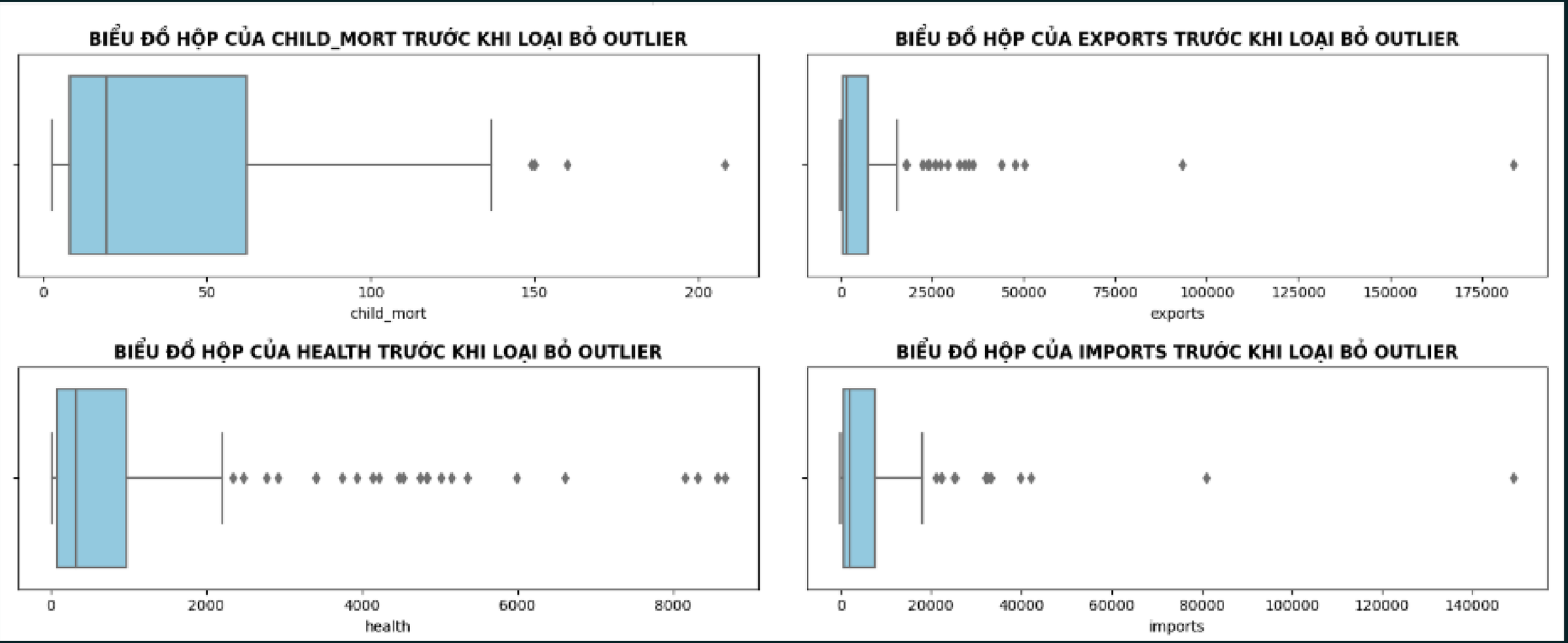
Thuộc tính	Mô tả
country	Tên quốc gia
child_mort	Tỷ lệ tử vong của trẻ em dưới 5 tuổi trên 1000 trẻ em
exports	Xuất khẩu hàng hóa và dịch vụ, tính theo % trên tổng GDP
health	Tổng chi tiêu y tế theo % tuổi trong Tổng GDP
imports	Nhập khẩu hàng hóa và dịch vụ. Tính theo % tuổi trong Tổng GDP
income	Thu nhập ròng mỗi người
inflation	Đo lường tốc độ tăng trưởng hàng năm của Tổng GDP
life_expec	Số năm trung bình mà một đứa trẻ mới sinh có thể sống được nếu mô hình tử vong hiện tại được giữ nguyên
total_fer	Số con mà mỗi phụ nữ sẽ sinh ra nếu tỷ suất sinh theo độ tuổi hiện tại không đổi
gdpp	GDP bình quân đầu người. Được tính bằng Tổng GDP chia cho tổng dân số.

Tổng quan bộ dữ liệu

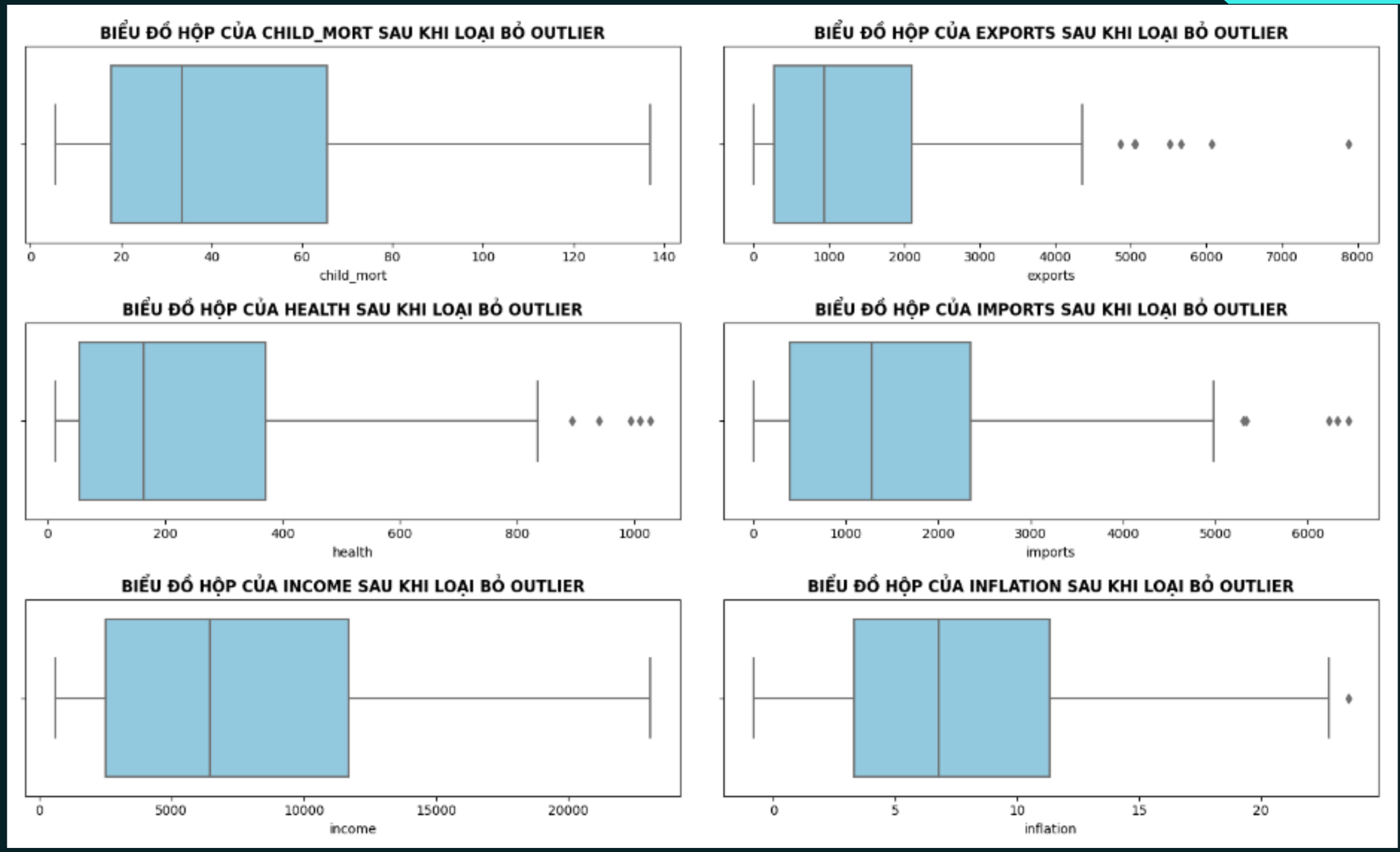
	country	child_mort	exports	health	imports	income	inflation	life_expec	total_fer	gdpp
0	Afghanistan	90.2	55.30	41.9174	248.297	1610	9.44	56.2	5.82	553
1	Albania	16.6	1145.20	267.8950	1987.740	9930	4.49	76.3	1.65	4090
2	Algeria	27.3	1712.64	185.9820	1400.440	12900	16.10	76.5	2.89	4460
3	Angola	119.0	2199.19	100.6050	1514.370	5900	22.40	60.1	6.16	3530
4	Antigua and Barbuda	10.3	5551.00	735.6600	7185.800	19100	1.44	76.8	2.13	12200

```
BỘ DỮ LIỆU COUNTRY TRƯỚC KHI XỬ LÝ MISSING VALUES
country       0
child_mort    0
exports       0
health        0
imports       0
income        0
inflation     0
life_expec    0
total_fer     0
gdpp          0
dtype: int64
```

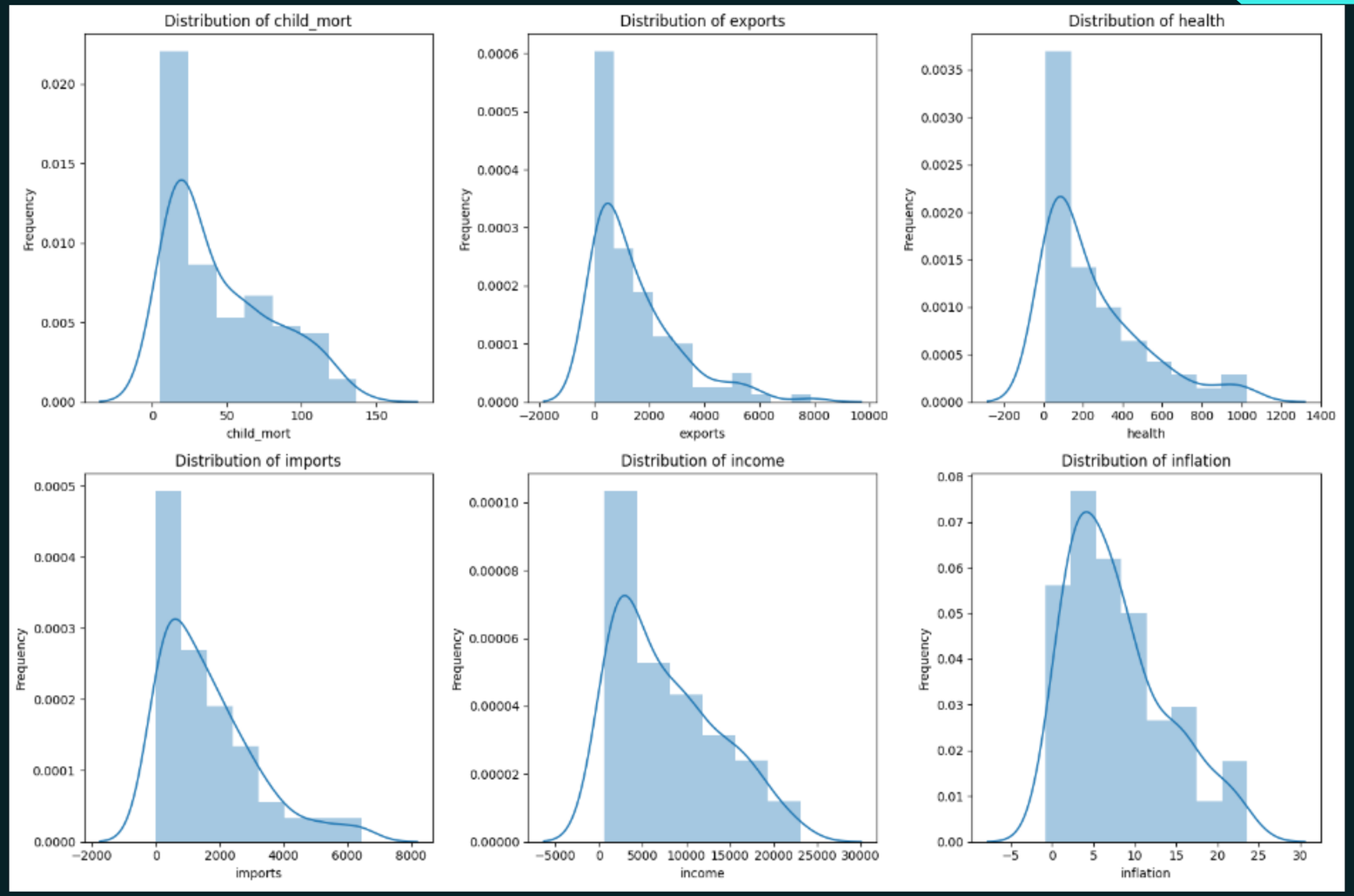
Tổng quan bộ dữ liệu



Tổng quan bộ dữ liệu



Tổng quan bộ dữ liệu



Tổng quan bộ dữ liệu

```
# Create a scaling object
scaler = StandardScaler()
# Create a list of the variables that you need to scale
varlist = ['child_mort', 'exports', 'health', 'imports', 'income',
           'inflation', 'life_expec', 'total_fer', 'gdpp']
# Scale these variables using 'fit_transform'
df[varlist] = scaler.fit_transform(df[varlist])
```

Chuẩn hóa dữ liệu định lượng bằng StandardScaler()

Tổng quan bộ dữ liệu

```
pca = IncrementalPCA(n_components=2)
# Putting feature variable to X
X = df.drop(['country'],axis=1)
# Putting response variable to y
y = df['country']
pca.fit(X)
```

```
IncrementalPCA(n_components=2)
```

Giảm chiều dữ liệu

	0	1
0	-3.261996	0.661249
1	1.278561	-1.302147
2	0.847099	0.528581
3	-2.154828	3.190433
4	3.083267	2.047718
...	...	...
106	-1.049480	-0.096031
107	-0.605183	-0.817434
108	-0.295497	-0.835275
109	-1.930999	1.433658
110	-2.910195	1.470137

111 rows × 2 columns

Áp dụng giải thuật

- Chạy giải thuật và trực quan phân cụm

Đối với thư viện pyclustering, các thuật toán phân nhóm yêu cầu tính toán khoảng cách giữa các điểm dữ liệu phải theo dạng số, nên phải xóa hoặc chuyển sang giá trị định lượng để phục vụ tiếp cho công việc chạy mô hình

```
data = df_pca.iloc[:,:].values
data
```

```
array([[-3.26199558,  0.66124889],
       [ 1.27856065, -1.30214717],
       [ 0.84709947,  0.5285812 ],
       [-2.15482828,  3.19043318],
       [ 3.08326668,  2.04771828],
       [ 0.35956019, -1.19054485],
       [ 1.51146601,  1.12415075],
       [-1.37125973, -1.31774923],
       [ 2.63950226,  0.9319004 ],
       [ 1.11133214, -0.99444023],
       [-2.95154199, -0.40260878],
       [-0.17403817, -1.0131912 ],
       [-0.80039719, -0.5952445 ],
```

Áp dụng giải thuật

__init__()

```
def pyclustering.cluster.bang.bang.__init__ (    self,
                                                 data,
                                                 levels,
                                                 ccore = False,
                                                 ** kwargs
                                                )
```

Create BANG clustering algorithm.

Parameters

[in] **data**	(list): Input data (list of points) that should be clustered.
[in] **levels**	(uint): Amount of levels in tree that is used for splitting (how many times block should be split). For example, if amount of levels is two then surface will be divided into two blocks and each obtained block will be divided into blocks also.
[in] **ccore**	(bool): Reserved positional argument - not used yet.
[in] ****kwargs**	Arbitrary keyword arguments (available arguments: 'observe').

```
# Prepare algorithm's parameters.
levels = 8

# Create instance of BANG algorithm.
bang_instance = bang(data, levels, metric='euclidean')
bang_instance.process()
```

Áp dụng giải thuật

- Để hoàn thành phân cụm cần sử dụng hàm process() để thực hiện giải thuật và trả các về các thông tin đầu ra.

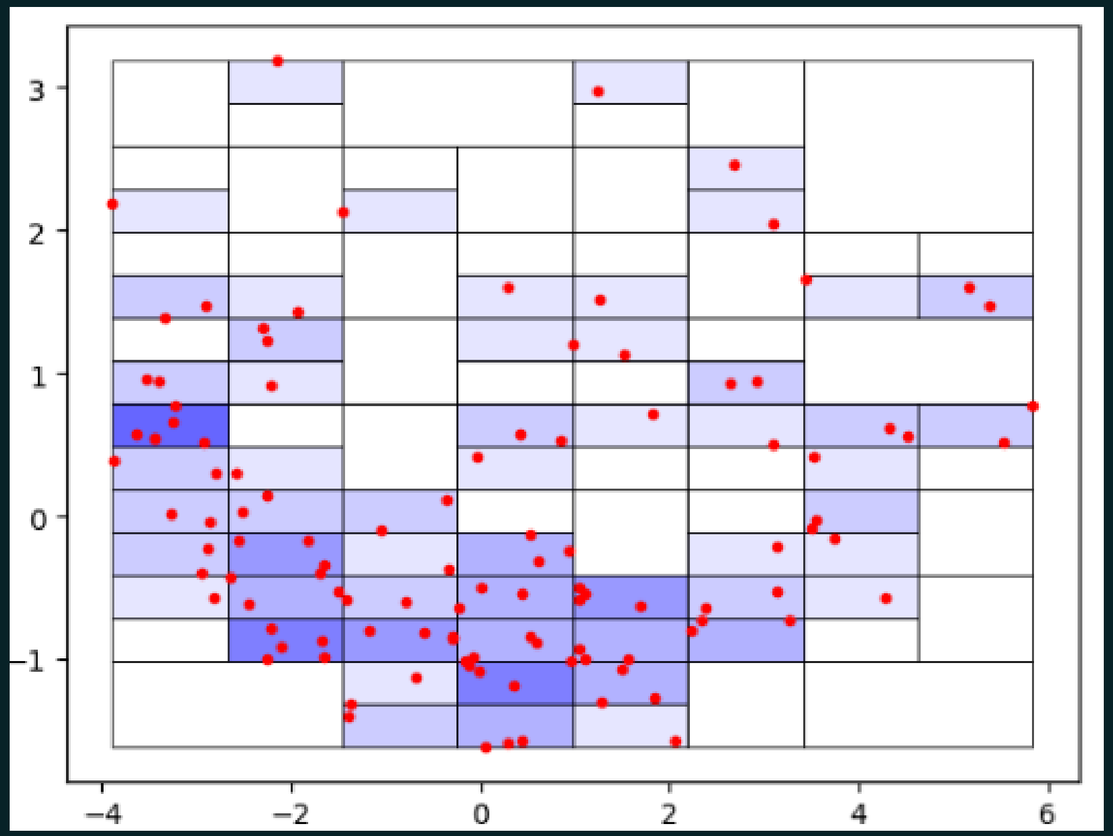
```
clusters = bang_instance.get_clusters()
noise = bang_instance.get_noise()
directory = bang_instance.get_directory()
dendrogram = bang_instance.get_dendrogram()
```

```
for i, cluster in enumerate(clusters):
    print(f'Cụm {i + 1}: Số lượng điểm {len(cluster)}')
```

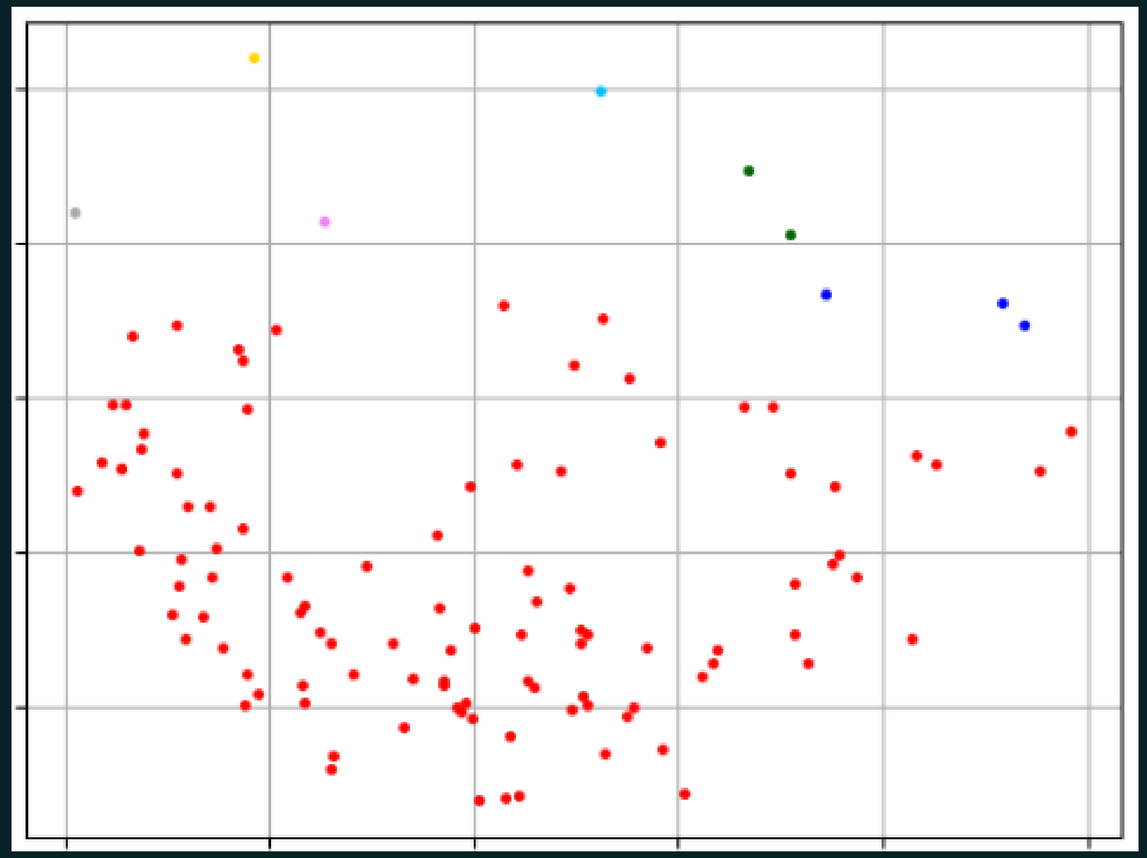
```
Cụm 1: Số lượng điểm 102
Cụm 2: Số lượng điểm 3
Cụm 3: Số lượng điểm 2
Cụm 4: Số lượng điểm 1
Cụm 5: Số lượng điểm 1
Cụm 6: Số lượng điểm 1
Cụm 7: Số lượng điểm 1
```

Áp dụng giải thuật

Để dễ hình dung hơn sự phân bố theo dạng lưới ô tiếp tục tiến hành việc vẽ những biểu đồ bằng các hàm thuộc lớp thư viện bang_visualizer().

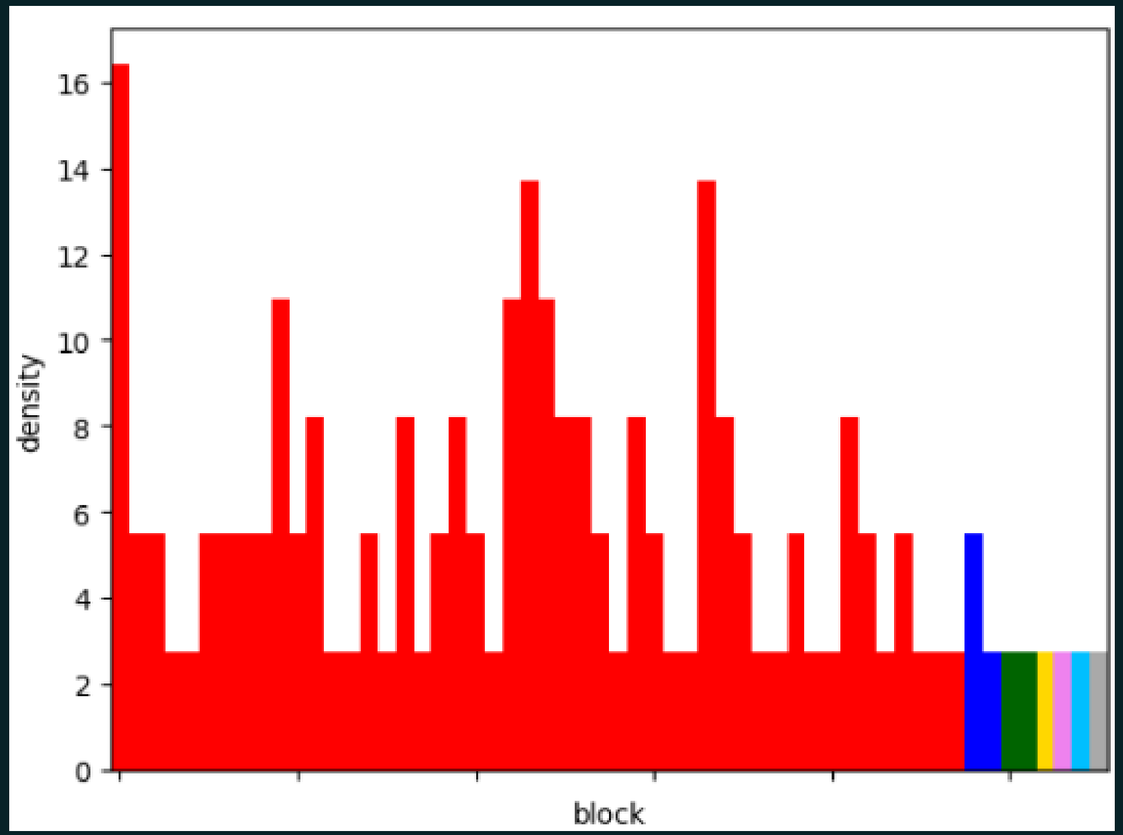


Áp dụng giải thuật



show_clusters(
)

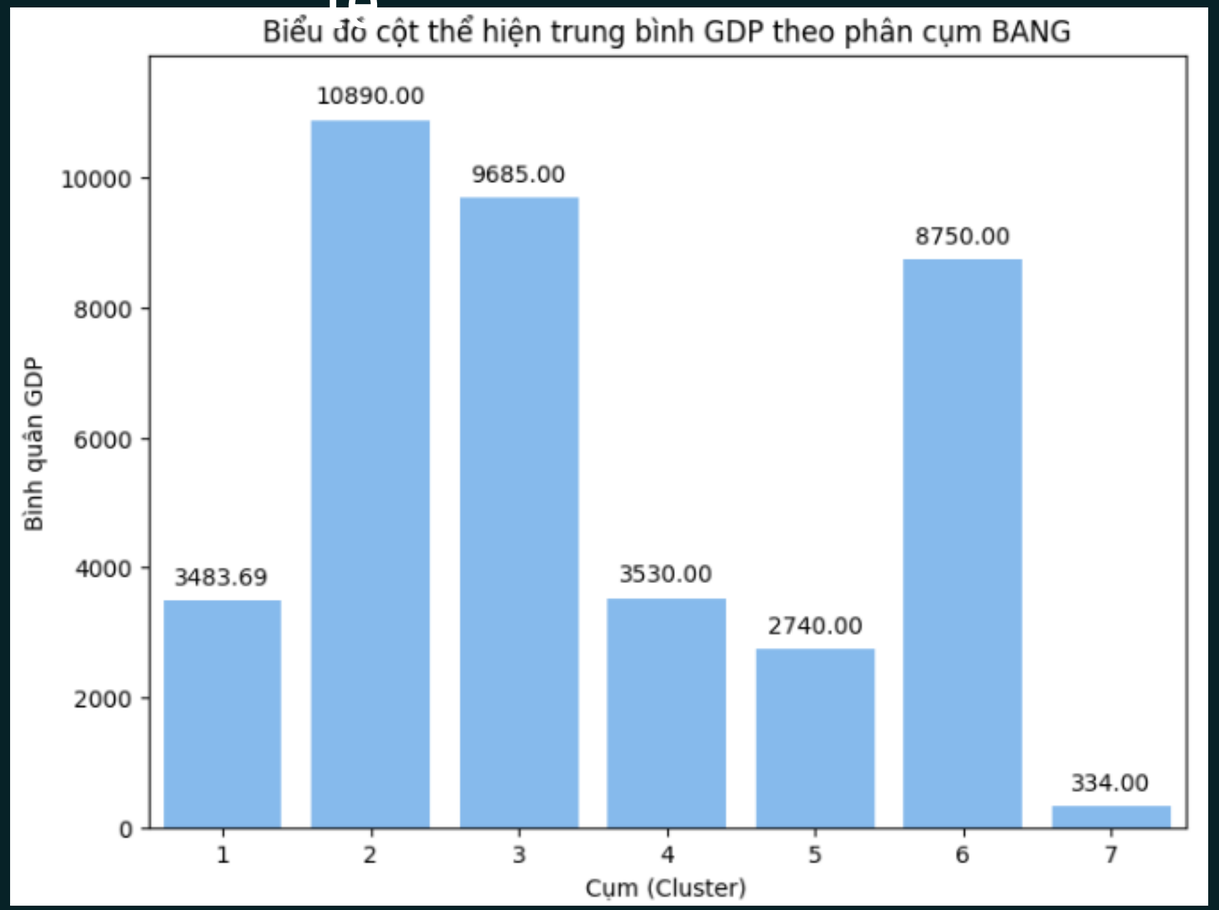
Áp dụng giải thuật

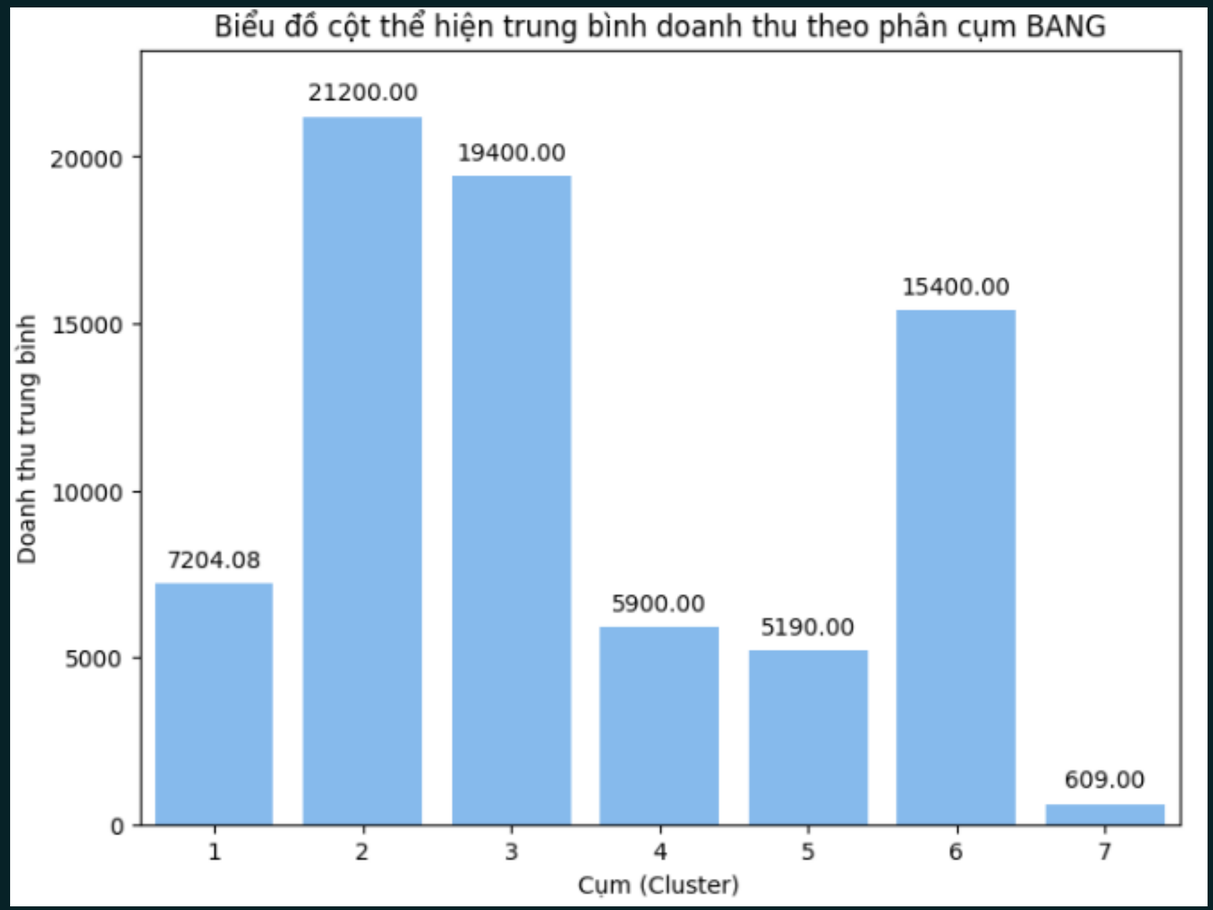


show_dendorgram()

Phân tích cụm cho bộ dữ liệu gốc dựa trên kết quả nhãn

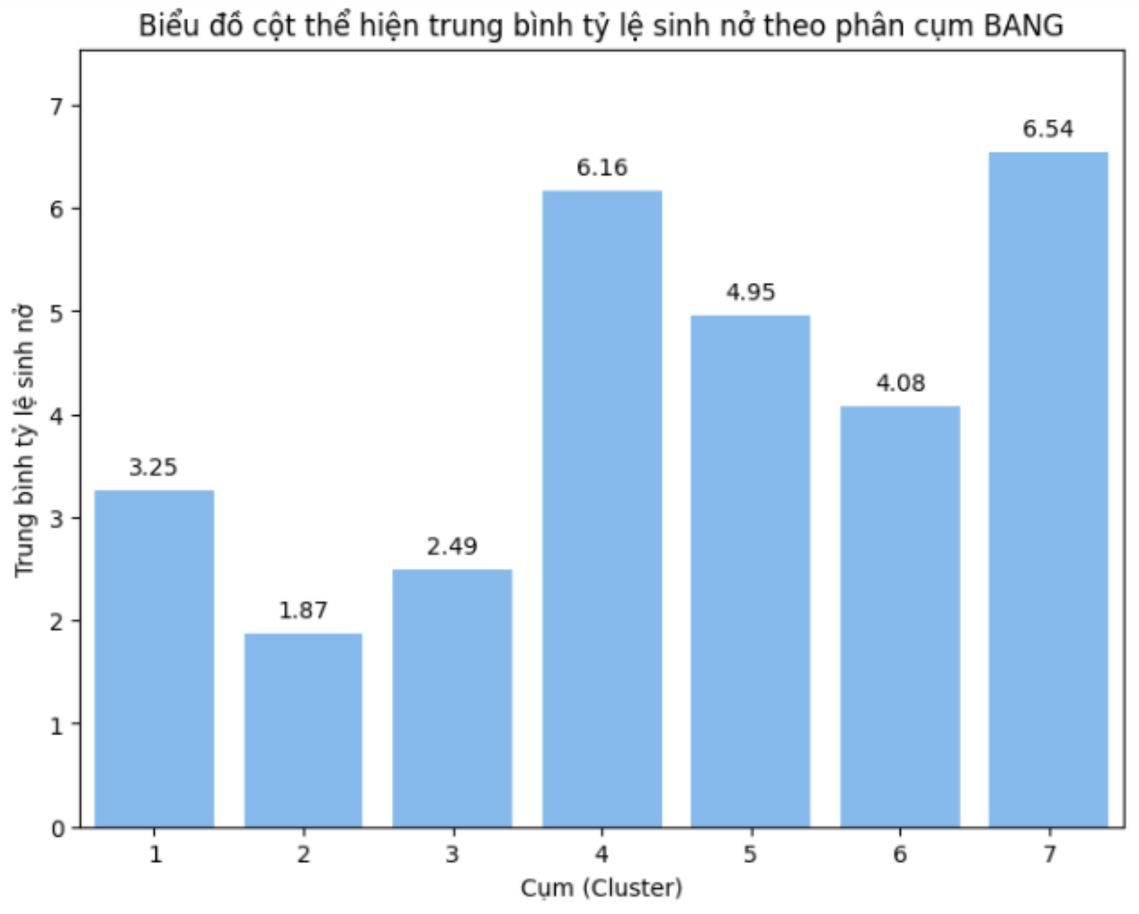
- Yếu tố kinh tế

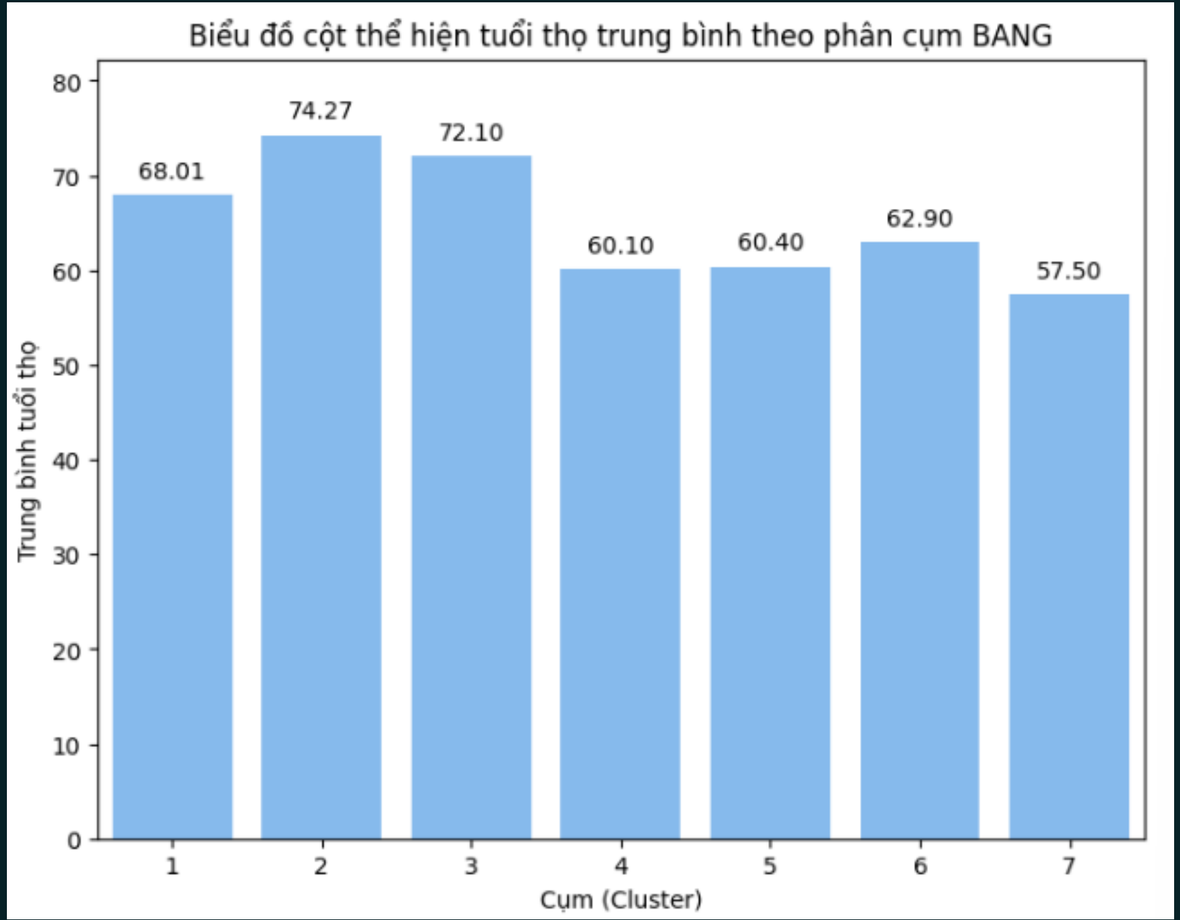




Phân tích cụm cho bộ dữ liệu gốc dựa trên kết quả nhãn

2. Yếu tố dân số

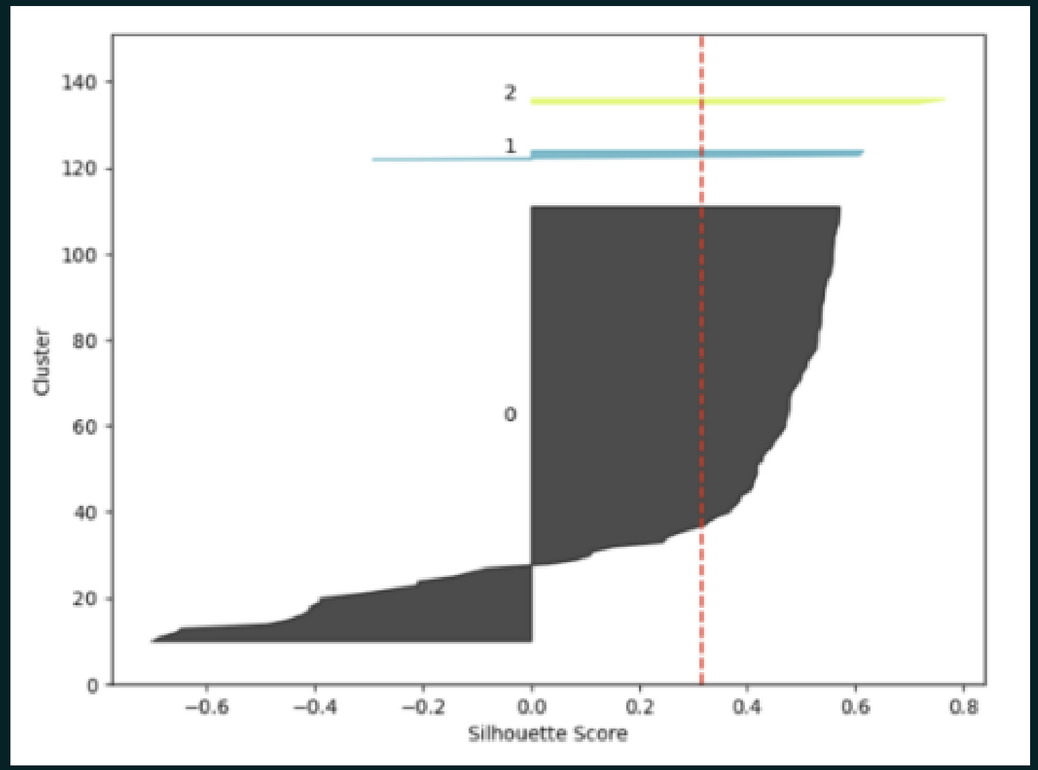




Đánh giá hiệu quả giải thuật

- Đánh giá chung

Cluster	Bigger	Smaller	Minus	Per > mean	Per < 0
0	73	29	22	71.56862745098039	21.568627450980394
1	2	1	1	66.66666666666666	33.33333333333333
2	2	0	0	100.0	0.0
all	77	30	23	71.96261682242991	21.49532710280374

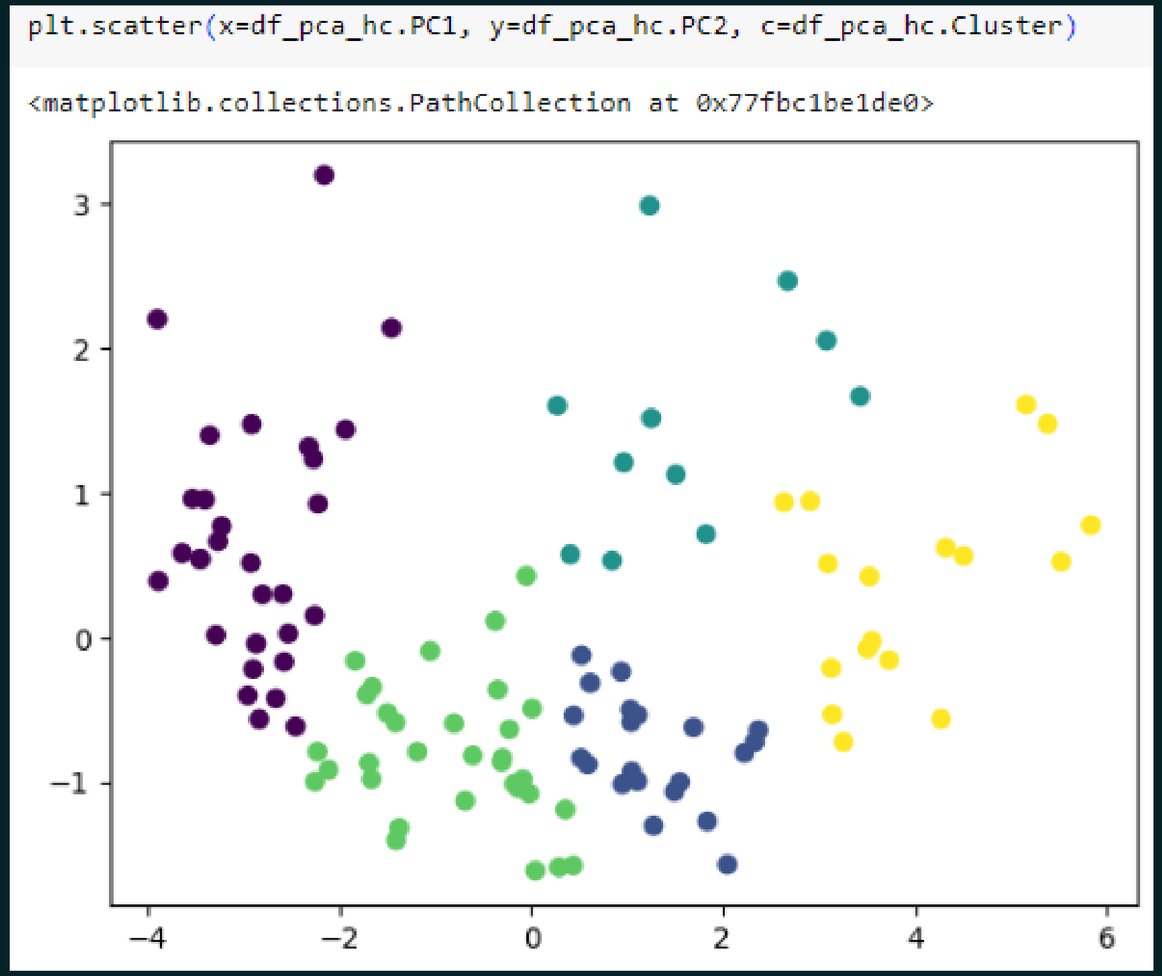


Đánh giá hiệu quả giải thuật

2. So sánh với thuật toán phân cụm HAC

```
plt.scatter(x=df_pca_hc.PC1, y=df_pca_hc.PC2, c=df_pca_hc.Cluster)
```

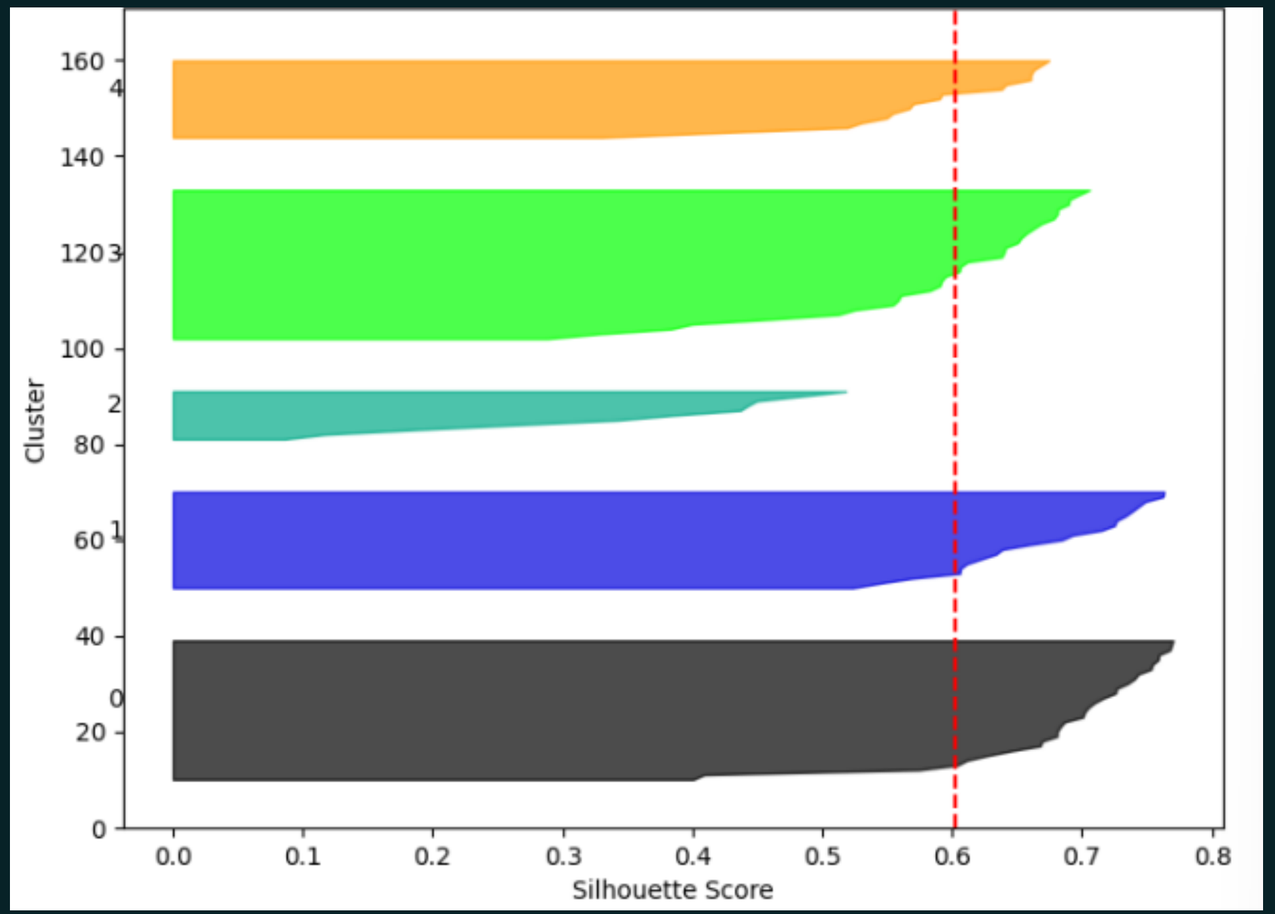
```
<matplotlib.collections.PathCollection at 0x77fbc1be1de0>
```



Đánh giá hiệu quả giải thuật

2. So sánh với thuật toán phân cụm HAC

Cluster	Bigger	Smaller	Minus	Per > mean	Per < 0
0	22	8	0	73.33333333333333	0.0
1	17	4	0	80.95238095238095	0.0
2	2	9	3	18.181818181818183	27.27272727272727
3	13	19	3	40.625	9.375
4	11	6	1	64.70588235294117	5.88235294117647
all	65	46	7	58.55855855855856	6.306306306306306

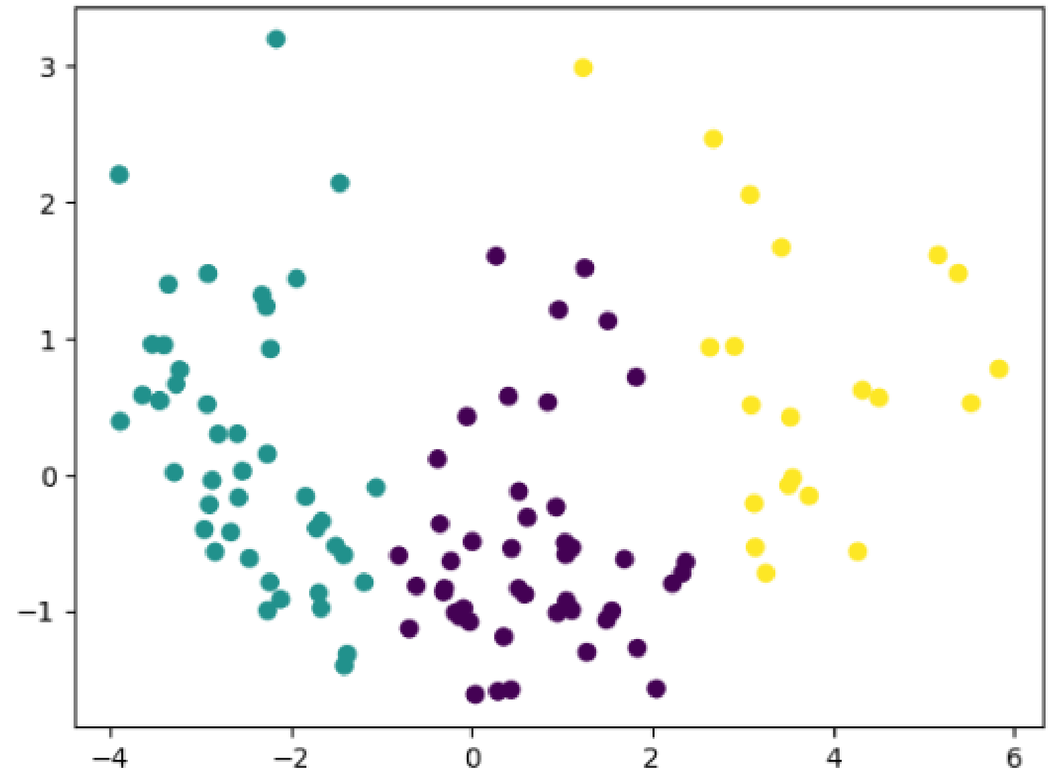


Đánh giá hiệu quả giải thuật

2. So sánh với thuật toán phân cụm K-Means

```
df_pca_kmean.columns = ['PC1', 'PC2', 'Cluster','SilhouetteScore' ]
plt.scatter(x=df_pca_kmean.PC1, y=df_pca_kmean.PC2, c=df_pca_kmean.Cluster)
```

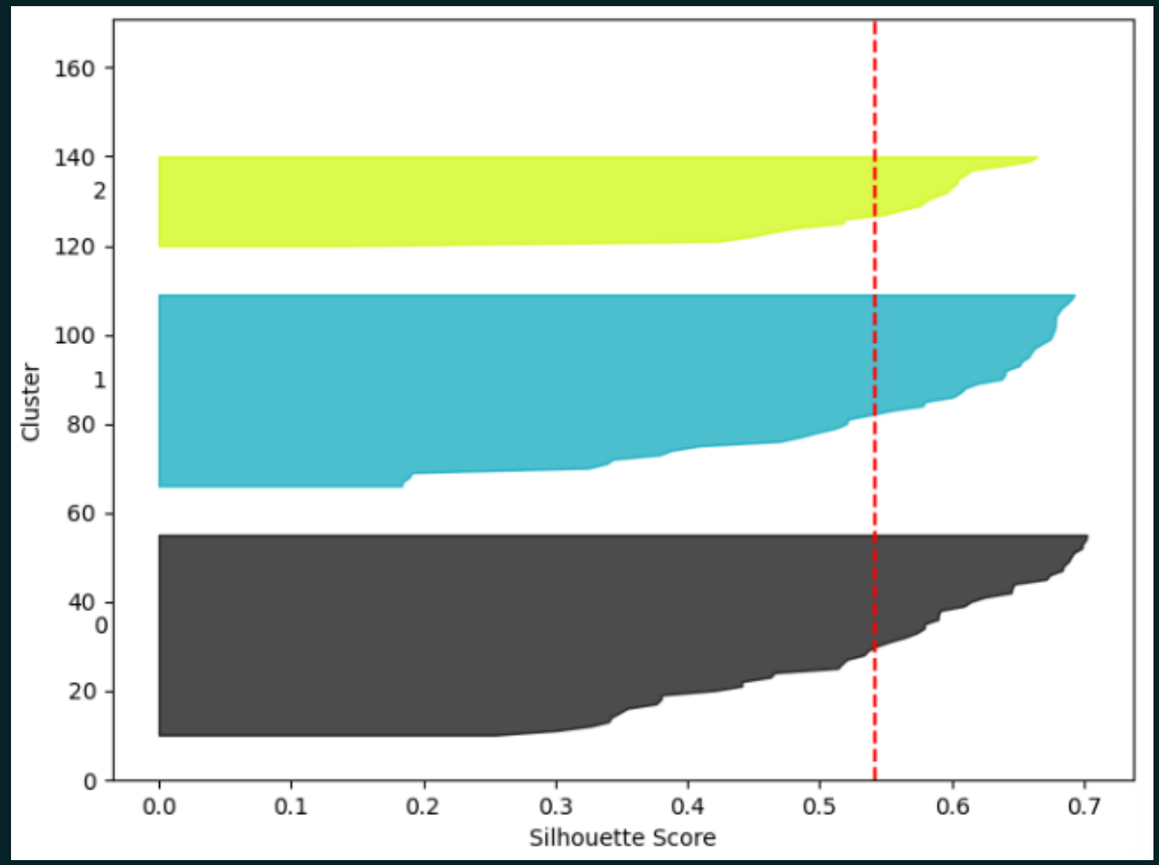
```
<matplotlib.collections.PathCollection at 0x77fbc1ed19c0>
```



Đánh giá hiệu quả giải thuật

2. So sánh với thuật toán phân cụm K-Means

Cluster	Bigger	Smaller	Minus	Per > mean	Per < 0
0	30	14	0	68.18181818181817	0.0
1	12	9	0	57.14285714285714	0.0
2	27	19	0	58.69565217391305	0.0
all	69	42	0	62.16216216216216	0.0



Đánh giá hiệu quả giải thuật

3. Độ phức tạp Big-O Notation

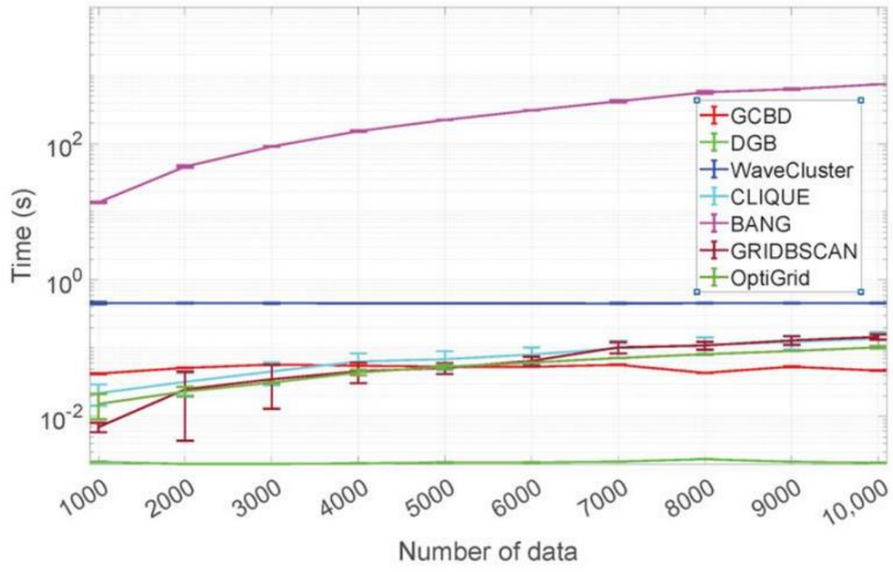
Bước 1: Khởi tạo cấu trúc lưới

Bước 2: Tính mật độ của các khối dữ liệu

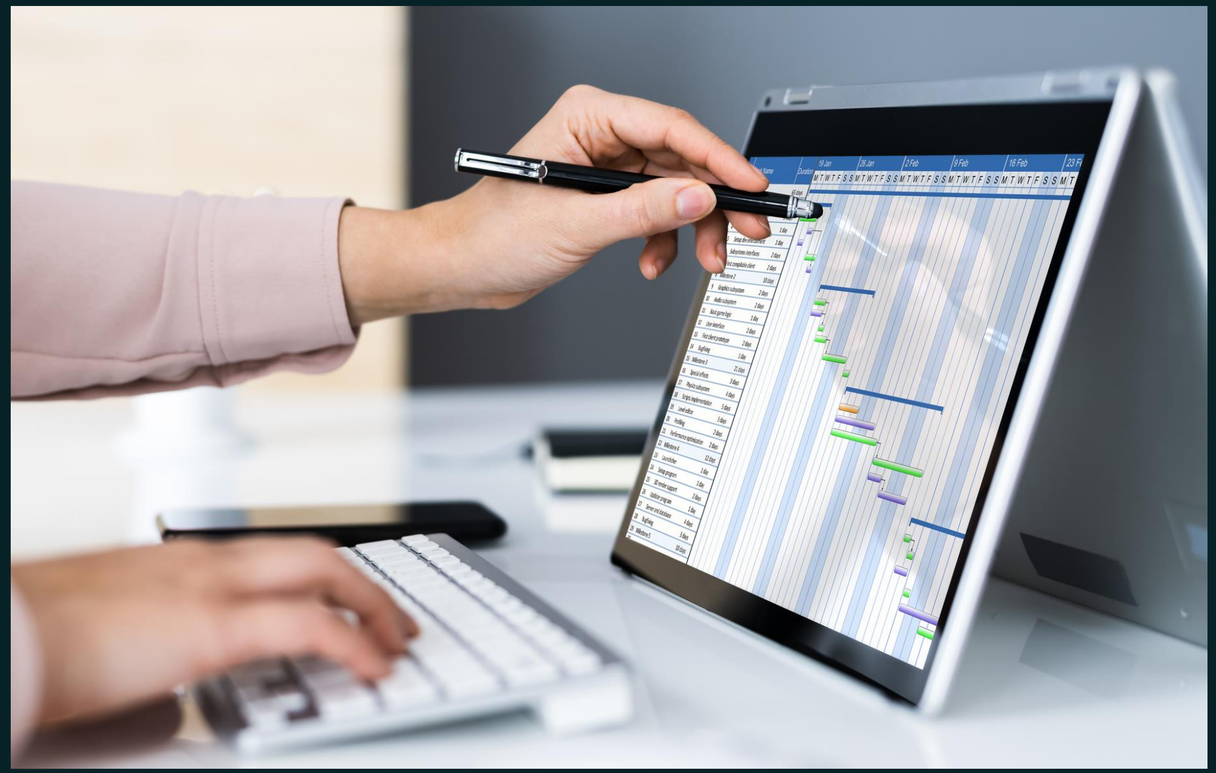
$O[n^2 log(n)]$

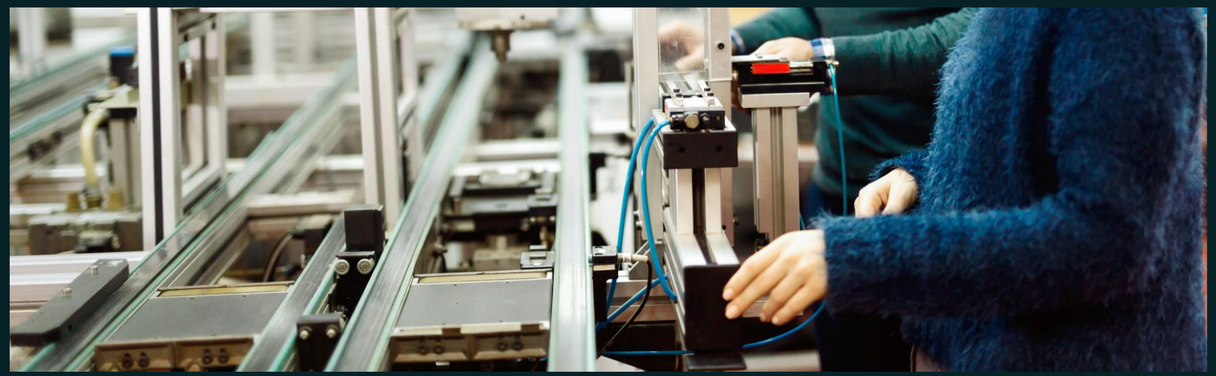
Names	#Instances	#Features	#Classes
Mickey	1200	2	3
Gu	1050	2	2
Jain	373	2	2
ThreeD	1300	2	3
DiffD	863	2	4
Moons	1000	2	2
Shape3	2250	2	3
Handl	715	2	3
Yinyang	3200	2	5
T4	7326	2	6
T7	9208	2	9
SF	16,384	2	4
ORL	100	10,307	10
Dermatology	366	34	6
Control	600	60	6
Dig	1797	64	10
Optdigits	5620	64	10
Satimage	6435	36	6

Hình X. Các tập dữ liệu được sử dụng trong thực nghiệm



Hình X. Biểu đồ đường so sánh thời gian chạy của các thuật toán khác nhau





Kết luận & Đánh giá

Bài đồ án đã khảo sát các chỉ số kinh tế-xã hội của các quốc gia trên thế giới và sử dụng thuật toán phân cụm BANG để phân loại chúng vào các nhóm có đặc điểm tương tự. Bài đồ án đã đạt được một số kết quả vì đã phân cụm các nước có tương đồng với nhau thành các nhóm

THANK YOU