

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



MÔN MẠNG XÃ HỘI

BÁO CÁO CUỐI KỲ

Đề tài: Accessing the mobility between countries base on “Air Social Networks”

Giáo viên hướng dẫn: Ths Thái Bảo Trân

Sinh viên thực hiện:

Trần Minh Ngọc – 20521669

Nguyễn Anh Nguyên – 20521674

Lê Thị Đoan Trang – 20522038

Nguyễn Thị Thảo Trang – 20522040

Tp. Hồ Chí Minh, 11/2023

MỤC LỤC

PHẦN 1: TỔNG QUAN	1
1. Giới thiệu đề tài	1
2. Giới thiệu dữ liệu	1
a. Nguồn dữ liệu	1
b. Mô tả dữ liệu	1
c. Xử lý và phân tích dữ liệu	3
d. Chuyển đổi data thành đồ thị	5
PHẦN 2: CÁC ĐỘ ĐO TRUNG TÂM	8
1. Degree Centrality	8
1.1 Sử dụng Python	8
1.2 Sử dụng Gephi	11
2. Closeness Centrality	11
2.1 Sử dụng Python	11
2.2 Sử dụng Gephi	15
3. Harmonic Centrality	15
3.1 Sử dụng Python	15
3.2 Sử dụng Gephi	18
4. Betweenness Centrality	19
4.1 Sử dụng Python	19
4.2 Sử dụng Gephi	22
5. Eigenvector Centrality	23
5.1 Sử dụng Python	23
5.2 Sử dụng Gephi	26
6. PageRank	26
6.1 Sử dụng Python	27
6.2 Sử dụng Gephi	29
PHẦN 3: CÁC THUẬT TOÁN GOM CỤM	30
1. Thuật toán Louvain	30

2. Thuật toán Girvan Newman	35
PHẦN 4: TÍNH TAY CÁC THUẬT TOÁN TRÊN 10 NODE NGẪU NHIÊN.....	39
1. Degree Centrality	40
2. Closeness Centrality	40
3. Harmonic Centrality	40
4. Betweenness Centrality	41
5. Eigenvector Centrality	41
6. PageRank	43
7. Thuật toán gom cụm Louvain	44
PHẦN 5: NGÔN NGỮ VÀ ỨNG DỤNG SẼ ÁP DỤNG	46
1. Ngôn ngữ	46
2. Ứng dụng	46
Tài liệu tham khảo	48

PHẦN 1: TỔNG QUAN

1. Giới thiệu đề tài

Trước và sau khi hứng chịu những ảnh hưởng của đại dịch Covid-19, thì chúng ta đang chứng kiến sự phát triển không ngừng của các phương tiện di chuyển và sự tăng, giảm giá đang dần trở nên phổ biến. Bên cạnh đó ngành hàng không chịu tác động mạnh mẽ. Bây giờ, chúng ta có thể lựa chọn từ một loạt các hãng hàng không khác nhau, và chúng đang cạnh tranh với nhau bằng cách đưa ra nhiều ưu đãi hấp dẫn để thu hút hành khách và tối ưu hóa lợi nhuận của họ.

Đề tài này mục tiêu khám phá cách các mạng xã hội trong ngành hàng không, hay "Air Social Networks," có thể cung cấp sự thú vị và hữu ích cho người dùng. Các Air Social Networks bao gồm thông tin về lịch trình chuyến bay, cập nhật tình hình sân bay, đánh giá của hành khách và thậm chí là cơ hội kết nối xã hội với những người bạn mới trên cùng một chuyến bay. Điều này mở ra một loạt các cơ hội để nghiên cứu cách mạng hóa việc di chuyển quốc tế và ứng dụng các công nghệ thông tin để tối ưu hóa trải nghiệm của khách hàng thông qua các thuật toán xếp hạng, phân cụm.

Input: Sử dụng tập dữ liệu ban đầu về các hãng hàng không từ nguồn dữ liệu Data World, sau khi đã được tiền xử lý.

Output: Đưa ra độ đo và xác định các cộng đồng hoặc mô hình mạng xã hội trong "Air Travel Network" để hỗ trợ việc phân tích và tối ưu hóa quy trình di chuyển bằng đường hàng không.

2. Giới thiệu dữ liệu

a. Nguồn dữ liệu

- Dữ liệu về top 15 hãng hàng không năm 2022: [Top 20 Best Airlines in the World, As Ranked by a Major Survey of Travelers \(businessinsider.com\)](https://www.businessinsider.com/top-20-best-airlines-in-the-world-2022)

- Dữ liệu về các quốc gia phủ sóng của các hãng hàng không: [flight-routes data on data.world | 2 datasets available](https://data.world/flight-routes)

→ bao gồm 3 file là Routes.csv, Airlines.csv, Airport.csv

b. Mô tả dữ liệu

Data được tổng hợp từ 3 file csv như sau:

- File Routes.csv: gồm dữ liệu về tuyến đi của các chuyến bay

	A	B	C	D	E	F	G	H
1	Airline	Airline ID	Source airport	Source airport ID	Destination airport	Destination airport ID	Codeshare	Stops
2	2B	410	AER	2965	KZN	2990		0 CR2
3	2B	410	ASF	2966	KZN	2990		0 CR2
4	2B	410	ASF	2966	MRV	2962		0 CR2
5	2B	410	CEK	2968	KZN	2990		0 CR2
6	2B	410	CEK	2968	OVV	4078		0 CR2
7	2B	410	DME	4029	KZN	2990		0 CR2
8	2B	410	DME	4029	NBC	6969		0 CR2
9	2B	410	DME	4029	TGK	\N		0 CR2
10	2B	410	DME	4029	UUA	6160		0 CR2
11	2B	410	EGO	6156	KGD	2952		0 CR2
12	2B	410	EGO	6156	KZN	2990		0 CR2

Hình: file Routes gốc

→ chọn ra 2 thuộc tính cần thiết là Airline ID (Mã hãng hàng không – màu vàng) và Destination airport (Mã IATA của sân bay mà hãng hàng không đó bay tới – màu xanh lá).

	A	B
1	Airline ID	Destination airport
2	410	KZN
3	410	KZN
4	410	MRV
5	410	KZN
6	410	OVV
7	410	KZN
8	410	NBC
9	410	TGK

Hình: File dữ liệu sau khi chọn ra 2 thuộc tính cần thiết

- File Airlines.csv:

	A	B	C	D	E	F	G	H
1	Airline ID	Name	Alias	IATA	ICAO	Callsign	Country	Active
2	-1	Unknown	\N	-		\N	\N	Y
3	1	Private flight	\N	-				Y
4	2	135 Airways	\N		GNL	GENERAL	United States	N
5	3	1Time Airline	\N	1T	RNX	NEXTIME	South Africa	Y
6	4	2 Sqn No 1 Elementary Flying Training School	\N		WYT		United Kingdom	N
7	5	213 Flight Unit	\N		TFU		Russia	N
8	6	223 Flight Unit State Airline	\N		CHD	CHKALOVSK-AVIA	Russia	N
9	7	224th Flight Unit	\N		TTF	CARGO UNIT	Russia	N
10	8	247 Jet Ltd	\N		TWF	CLOUD RUNNER	United Kingdom	N
11	9	2D Aviation	\N		SEC	SECUREV	United States	N

Hình: File Airlines gốc

→ Cột Airline ID (màu xanh lá) dùng để làm khóa ngoại với cột Airline ID trong file Routes.csv, cột Name (màu vàng) là dữ liệu sẽ được lấy ra.

- File Airports.csv:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Airport ID	Name	City	Country	IATA	ICAO	Latitude	Longitude	Altitude	Timezone	DST	Tz database time zone	Type	Source
1	Goroka Airport	Goroka	Papua New Guinea	GKA	AYGA	-6.0817	145.392	5282	10 U		Pacific/Port_Moresby	airport	OurAirports
2	Madang Airport	Madang	Papua New Guinea	MAG	AYMD	-5.2071	145.789	20	10 U		Pacific/Port_Moresby	airport	OurAirports
3	Mount Hagen Kagamuga Airport	Mount Hagen	Papua New Guinea	HGU	AYMH	-5.8268	144.29601	5388	10 U		Pacific/Port_Moresby	airport	OurAirports
4	Nadzab Airport	Nadzab	Papua New Guinea	LAE	AYNZ	-6.5698	146.72598	239	10 U		Pacific/Port_Moresby	airport	OurAirports
5	Port Moresby Jacksons International Airport	Port Moresby	Papua New Guinea	POM	AYPY	-9.4434	147.22	146	10 U		Pacific/Port_Moresby	airport	OurAirports
6	Wewak International Airport	Wewak	Papua New Guinea	WWK	AYWK	-3.5838	143.66901	19	10 U		Pacific/Port_Moresby	airport	OurAirports
7	Narsarsuaq Airport	Narsarsuaq	Greenland	UAK	BGBW	61.1605	-45.426	112	-3 E		America/Godthab	airport	OurAirports
8	Godthaab / Nuuk Airport	Godthaab	Greenland	GOH	BGGH	64.1909	-51.6781	283	-3 E		America/Godthab	airport	OurAirports
9	Kangerlussuaq Airport	Sondrestrom	Greenland	SFJ	BGSF	67.0122	-50.7116	165	-3 E		America/Godthab	airport	OurAirports
10	Thule Air Base	Thule	Greenland	THU	BGTL	76.5312	-68.7032	251	-4 E		America/Thule	airport	OurAirports
11	Reykjavik Airport	Reykjavik	Iceland	RFV	RIAR	65.66	-18.0777	6	0 N		Atlantic/Reykjavik	airport	OurAirports

Hình: File Airports gốc

→ Cột IATA (màu xanh lá) dùng để làm khóa ngoại với cột Destination airport trong file Routes.csv, cột Country (màu vàng) là dữ liệu sẽ được lấy ra.

- Sau khi kết 3 file từ data như trên, ta được file Airline.csv chứa dữ liệu cuối cùng như sau:

	A	B
1	Airline	DestinationCountry
2	Air France	United States
3	Air France	Cote d'Ivoire
4	Air France	United States
5	Air France	Nigeria
6	Air France	Nigeria
7	Air France	United States
8	Air France	United Kingdom
9	Air France	United States
10	Air France	Spain

Hình: File Airline.csv

Bộ dữ liệu này chứa thông tin về các hãng hàng không và tên quốc gia mà các hãng hàng không đó bay đến.

Bảng thuộc tính:

STT	Thuộc tính	Kiểu dữ liệu	Mô tả	Các giá trị
1	Airline	String	Tên hãng hàng không	AirFrance,...
2	Country	String	Tên quốc gia mà hãng bay đi đến	United State,...

Bảng: Mô tả các trường dữ liệu

c. Xử lý và phân tích dữ liệu

i. Làm sạch dữ liệu

a) Import các thư viện cần thiết

```
[ ] import pandas as pd
import networkx as nx
from networkx.algorithms import bipartite
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.cm as cm
import matplotlib
from community import community_louvain
import seaborn as sns
from termcolor import colored
from networkx.algorithms.community import label_propagation_communities
from sklearn import preprocessing
from sklearn.cluster import KMeans
```

Hình: Các thư viện được import

b) Thực hiện làm sạch dữ liệu

```
#đọc dữ liệu từ file csv
dt=pd.read_csv("airline.csv", encoding="ISO-8859-1")
#xóa các dòng dữ liệu null
dt.dropna()
#Xóa các dòng dữ liệu bị trùng lặp
dt.drop_duplicates()
dt
```

Hình: Đoạn code làm sạch dữ liệu file Airline.csv

- Kết quả thu được:

	Airline	DestinationCountry
0	Air France	United States
1	Air France	Cote d'Ivoire
2	Air France	United States
3	Air France	Nigeria
4	Air France	Nigeria
...
6135	Virgin Atlantic Airlines	United States
6136	Virgin Atlantic Airlines	Saint Lucia
6137	Virgin Atlantic Airlines	Saint Lucia
6138	Virgin Atlantic Airlines	Saint Lucia
6139	Virgin Atlantic Airlines	Russia

6140 rows x 2 columns

Hình: Kết quả thu được sau khi làm sạch data

- Xuất đồ thị 1 phía ra file csv để chuẩn bị dữ liệu thực hiện trên phần mềm Gephi

```
labels = nx.get_edge_attributes(G, 'weight')
df2 = pd.DataFrame(columns=['source', 'target', 'weight'])
for key, value in labels.items():
    df2 = df2.append({'source':key[0], 'target':key[1], 'weight':value}, ignore_index=True)
df2.to_csv (r'gelphi.csv', index = False, header=True)
```

Hình: Đoạn code xuất đồ thị 1 phía ra file csv

- Kết quả thu được:

	A	B	C
1	source	target	weight
2	United States	Cyprus	7
3	United States	French Polynesia	4
4	United States	Jordan	7
5	United States	Swaziland	1
6	United States	Belgium	6
7	United States	Slovenia	2
8	United States	French Guiana	1
9	United States	Romania	6
10	United States	Portugal	7

Hình: File csv sau khi xuất

b) Đồ thị hai phía

- Chú thích đồ thị:
 - + Node: là tên của hàng không và tên các quốc gia phủ sóng của nó.
 - + Edge: mối quan hệ giữa các quốc gia của các hãng hàng không.
- Code:

ĐỒ THỊ 2 PHÍA

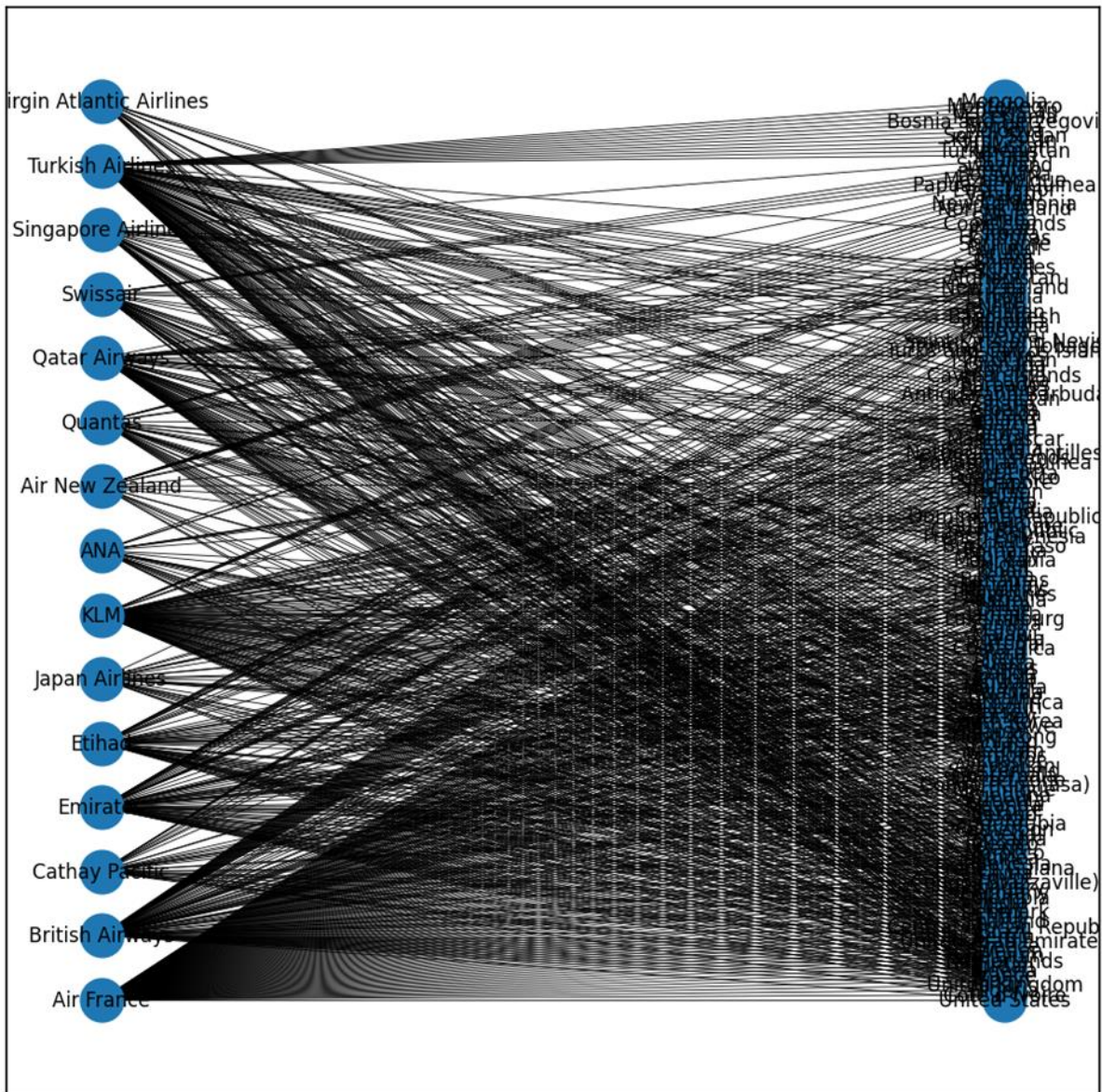
```
[4] for index, row in dt.iterrows():
    # Thêm cạnh vào đồ thị B, cạnh nối giữa hãng hàng không ('Airline') và quốc gia đích ('DestinationCountry')
    # Trọng số của cạnh được đặt là 1
    B.add_edge(row['Airline'], row['DestinationCountry'], weight=1)

    # Thêm các đỉnh vào đồ thị B cho tập hợp các quốc gia đích ('DestinationCountry')
    B.add_nodes_from(DestinationCountry,bipartite=0)
    # Thêm các đỉnh vào đồ thị B cho tập hợp các hãng hàng không ('Airline')
    B.add_nodes_from(Airline,bipartite=1)

plt.figure(figsize=(12,12))
pos = nx.spring_layout(B)
fig, ax = plt.subplots(1,1,figsize=(8,8), dpi=150)
nx.draw_networkx(B, pos = nx.drawing.layout.bipartite_layout(B, Airline),font_size=8,width=0.4)
```

Hình: Code hiển thị đồ thị 2 phía

- Kết quả:



PHẦN 2: CÁC ĐỘ ĐO TRUNG TÂM

1. Degree Centrality

Degree Centrality là một thước đo trung tâm đơn giản để đếm xem một nút có bao nhiêu nút lân cận. Nếu mạng được định hướng thì sẽ có hai phiên bản của thước đo: in-degree là số lượng liên kết đến hoặc số lượng nút tiền thân; out-degree là số lượng liên kết bên ngoài hoặc số lượng nút kế thừa.

Ý nghĩa: những quốc gia có degree centrality cao là những quốc gia có nhiều đường bay và nhiều liên kết đến các quốc gia khác, có vai trò quan trọng trong mạng lưới hàng không.

1.1 Sử dụng Python

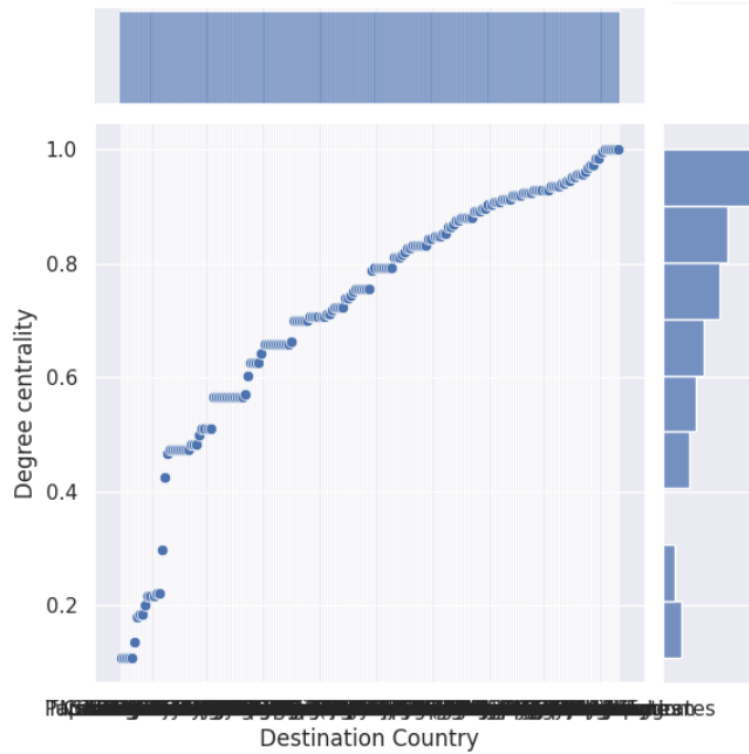
```
deg_cen = nx.degree_centrality(G) # return a dictionary
deg_cen_df = print_table(deg_cen.items(), ['Destination Country',
'Degree centrality'])
print("Degree Centrality of each country \n")
deg_cen_df
```

Degree Centrality of each country

	Destination Country	Degree centrality
0	Bangladesh	0.6413043478
1	Switzerland	0.9728260870
2	South Sudan	0.5652173913
3	Ethiopia	0.7880434783
4	Malta	0.8804347826
...
180	Niue	0.1086956522
181	Samoa	0.1086956522
182	Norfolk Island	0.1086956522
183	Cook Islands	0.1086956522
184	Tonga	0.1086956522

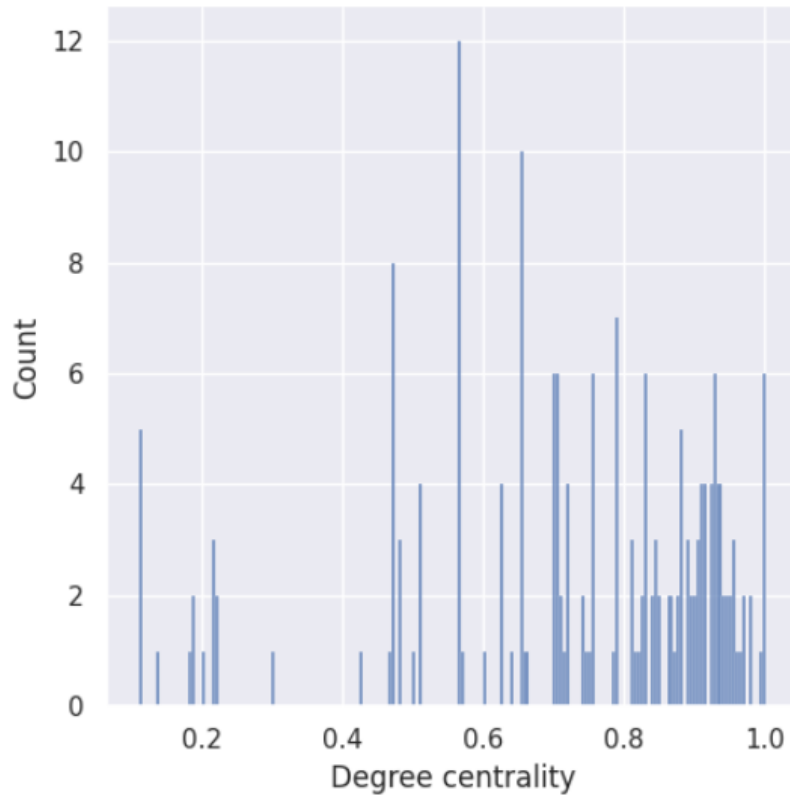
Hình 1: Kết quả tính giá trị Degree Centrality

```
print(colored("Biểu đồ thể hiện giá trị bậc của các node trong
graph", 'green', attrs=['bold']))
sns.jointplot(x="Destination Country", y="Degree centrality",
data=deg_cen_df)
```



Hình 2: Biểu đồ thể hiện giá trị bậc của các node

```
print(colored("Biểu đồ thể hiện phân bố Degree Centrality của các  
node trong graph", 'green', attrs=['bold']))  
sns.displot(x="Degree centrality", data=deg_cen_df, binwidth=0.005)
```



Hình 3: Biểu đồ thể hiện sự phân bố Degree Centrality của các node

```
print('Top 6 quốc gia có Degree centrality cao nhất: ')
deg_cen_df = deg_cen_df.sort_values(by='Degree centrality',
ascending=False)
deg_cen_df.head(6)
```

Top 6 quốc gia có Degree centrality cao nhất:

	Destination Country	Degree centrality
52	Singapore	1.0
51	Hong Kong	1.0
113	Germany	1.0
39	United Arab Emirates	1.0
114	United Kingdom	1.0
67	United States	1.0

Hình 4: Liệt kê các quốc gia có giá trị Degree cao nhất

```
print('Top 5 quốc gia có Degree centrality thấp nhất: ')
deg_cen_df = deg_cen_df.sort_values(by='Degree centrality',
ascending=True)
```

```
deg_cen_df.head(5)
```

Top 5 quốc gia có Degree centrality thấp nhất:

	Destination Country	Degree centrality
184	Tonga	0.1086956522
180	Niue	0.1086956522
183	Cook Islands	0.1086956522
182	Norfolk Island	0.1086956522
181	Samoa	0.1086956522

Hình 5: Liệt kê các quốc gia có giá trị Degree thấp nhất

1.2 Sử dụng Gephi

Id	destination country	degree centrality ▾
179	Germany	1.0
181	Hong Kong	1.0
184	Singapore	1.0
180	United Arab Emirates	1.0
183	United Kingdom	1.0
182	United States	1.0

Hình 6: Các quốc gia có giá trị Degree cao nhất

Id	destination country	degree centrality ^
2	Cook Islands	0.10869565217391304
1	Niue	0.10869565217391304
3	Norfolk Island	0.10869565217391304
4	Samoa	0.10869565217391304
0	Tonga	0.10869565217391304

Hình 7: Các quốc gia có giá trị Degree thấp nhất

2. Closeness Centrality

Closeness Centrality là độ đo cho biết độ "gần gũi" của mỗi node trong mạng đối với các node khác. Nếu giá trị của một node thấp, thì nó cần nhiều bước để kết nối với các node khác, làm cho việc giao tiếp trở nên phức tạp. Điều này giúp xác định những node trung tâm và quan trọng trong mạng.

Ý nghĩa: những quốc gia có closeness centrality cao là những quốc gia được phủ sóng bởi nhiều hãng hàng không khác nhau.

2.1 Sử dụng Python

```
closeness_centrality = nx.closeness_centrality(G)
```

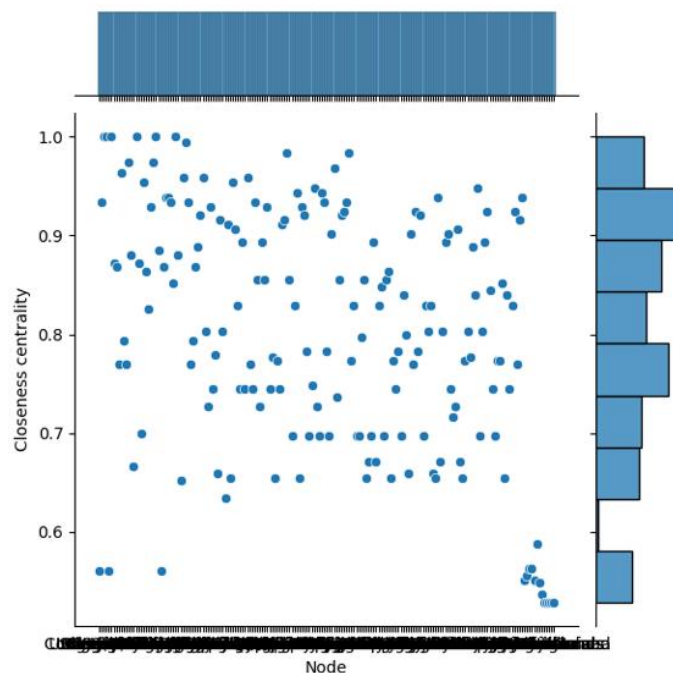
```
sorted_closeness centrality = sorted(closeness centrality.items(),
key=lambda x:x[1], reverse=True)
closeness centrality_df = print_table(closeness centrality.items(),
['Node', 'Closeness centrality'])
closeness centrality_df
```

	Node	Closeness centrality
0	Canada	0.968421
1	Japan	0.983957
2	Macau	0.536443
3	New Caledonia	0.562691
4	Iran	0.727273
...
180	Brunei	0.555891
181	East Timor	0.550898
182	Lesotho	0.560976
183	Swaziland	0.560976
184	Botswana	0.560976

Hình 1: Kết quả tính giá trị Closeness Centrality

```
print(colored("Biểu đồ thể hiện giá trị closeness centrality của các
node trong graph", 'green', attrs=['bold']))
```

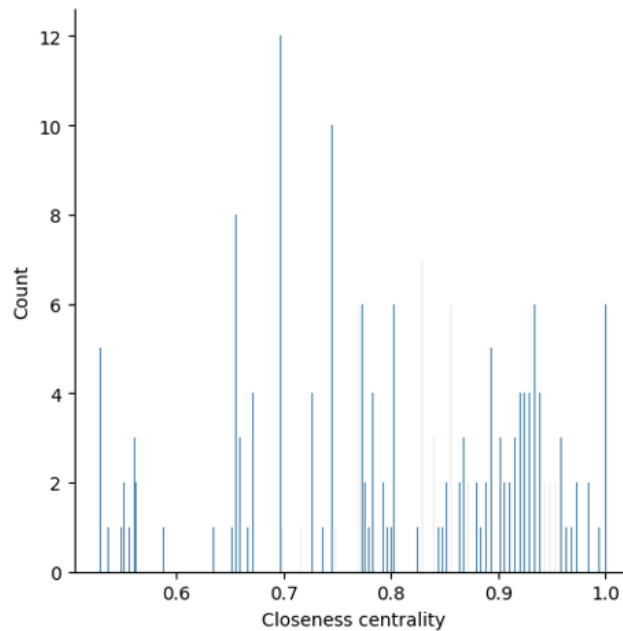
```
sns.jointplot(x="Node",y="Closeness centrality",
data=closeness centrality_df)
```



Hình 2: Biểu đồ thể hiện giá trị bậc của các node

```
print(colored("Biểu đồ thể hiện phân bố closeness centrality của các  
node trong graph", 'green', attrs=['bold']))
```

```
sns.displot(x="Closeness centrality", data=closeness_centrality_df,  
binwidth=0.001)
```



Hình 3: Biểu đồ thể hiện sự phân bố Closeness Centrality của các node

```
top_ten_max_closeness_cen = sorted_closeness_centrality[:10]
```

```
print('Top 10 quốc gia có Closeness centrality cao nhất: ')
```

```
print_table(top_ten_max_closeness_cen, ['Country', 'Closeness  
centrality'])
```


Top 10 quốc gia có Closeness centrality cao nhất:

	Counrty	Closeness centrality
0	Germany	1.000000
1	Singapore	1.000000
2	United States	1.000000
3	United Arab Emirates	1.000000
4	Hong Kong	1.000000
5	United Kingdom	1.000000
6	Australia	0.994595
7	China	0.983957
8	Japan	0.983957
9	Switzerland	0.973545

Hình 4: Liệt kê các quốc gia có giá trị Closeness cao nhất

```
top_ten_min_closeness_cen = sorted_closeness_centrality[-10:]
print('Top 10 quốc gia có Closeness centrality nhỏ nhất: ')
print_table(top_ten_min_closeness_cen, ['Counrty', 'Closeness
centrality'])
```

Top 10 quốc gia có Closeness centrality nhỏ nhất:

	Counrty	Closeness centrality
0	Brunei	0.555891
1	East Timor	0.550898
2	Papua New Guinea	0.550898
3	Guam	0.549254
4	Macau	0.536443
5	Tonga	0.528736
6	Niue	0.528736
7	Samoa	0.528736
8	Cook Islands	0.528736
9	Norfolk Island	0.528736

Hình 5: Liệt kê các quốc gia có giá trị Closeness nhỏ nhất

2.2 Sử dụng Gephi

Id	Closeness Centrality ▾
United States	1.0
United Arab Emira...	1.0
United Kingdom	1.0
Germany	1.0
Hong Kong	1.0
Singapore	1.0
Australia	0.994595
China	0.983957
Japan	0.983957
Thailand	0.9735449735449735

Hình 6: Top 10 quốc gia có Closeness centrality cao nhất

Id	Closeness Centrality ^
Niue	0.528736
Norfolk Island	0.528736
Samoa	0.528736
Tonga	0.528736
Cook Islands	0.528736
Macau	0.536443
Guam	0.549254
Papua New Guinea	0.550898
East Timor	0.550898
Brunei	0.5558912386706949

Hình 7: Top 10 quốc gia có Closeness centrality nhỏ nhất

3. Harmonic Centrality

Harmonic Centrality còn được gọi là độ đo trung tâm có giá trị là một phần mở rộng của độ đo trung tâm gần, là một độ đo về tầm quan trọng của các đỉnh (nodes) trong mạng (graph) dựa trên độ dài của đường đi ngắn nhất từ một đỉnh đến tất cả các đỉnh khác trong mạng. Là một phát minh để giải quyết vấn đề mà công thức ban đầu gặp phải khi xử lý các đồ thị không liên kết.

Ý nghĩa: các quốc gia có giá trị harmonic centrality cao có thể chỉ ra rằng có nhiều hãng hàng không đi đến quốc gia đó và từ quốc gia đó hành khách có thể có nhiều sự lựa chọn về hãng hàng không và các dịch vụ hàng không khác nhau.

3.1 Sử dụng Python

```

harmonic_cen = nx.harmonic_centralty(G)

# Sắp xếp giảm dần các giá trị harmonic centrality và lưu vào sorted_closeness centrality
sorted_closeness centrality = sorted(harmonic_cen.items(), key=lambda x:x[1], reverse=True)

# Tạo DataFrame hiển thị thông tin harmonic centrality cho từng đỉnh
harmonic_cen_df = print_table(harmonic_cen.items(), ['Node', 'Harmonic centrality'])

harmonic_cen_df

```

	Node	Harmonic centrality
0	Palau	153.0
1	Saint Kitts and Nevis	135.5
2	Panama	157.0
3	Italy	177.0
4	Sierra Leone	165.0
...
180	Bahamas	161.5
181	United Kingdom	184.0
182	Japan	182.5
183	Peru	157.5
184	Albania	156.5

185 rows x 2 columns

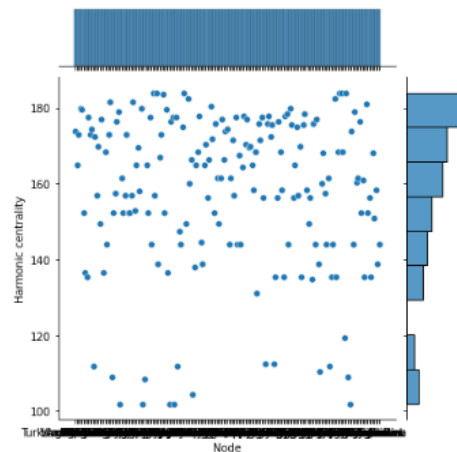
Hình 1: Kết quả tính giá trị Harmonic Centrality

```

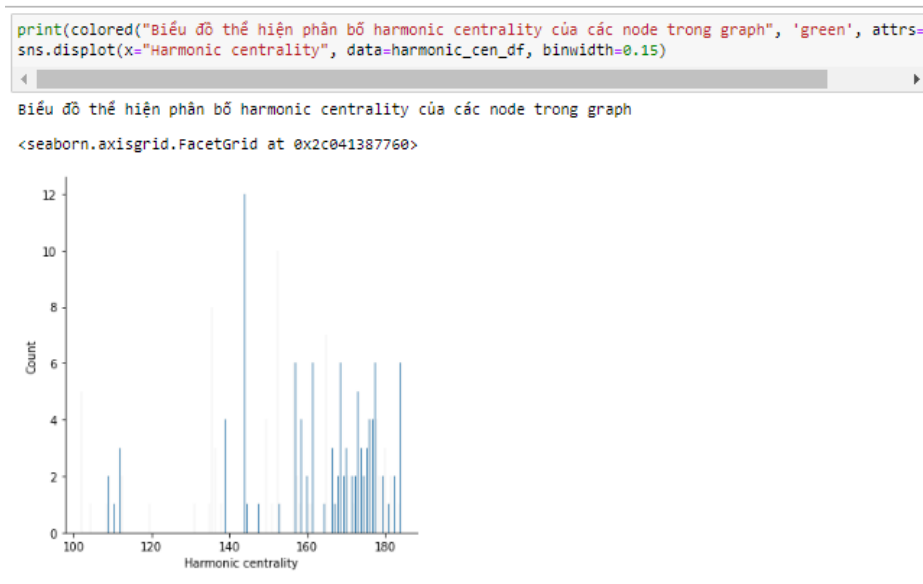
print(colored("Biểu đồ thể hiện giá trị harmonic centrality của các node trong graph", 'green', attrs=
sns.jointplot(x="Node", y="Harmonic centrality", data=harmonic_cen_df)

```

Biểu đồ thể hiện giá trị harmonic centrality của các node trong graph
<seaborn.axisgrid.JointGrid at 0x2c03fc54640>



Hình 2: Biểu đồ thể hiện giá trị bậc của các node



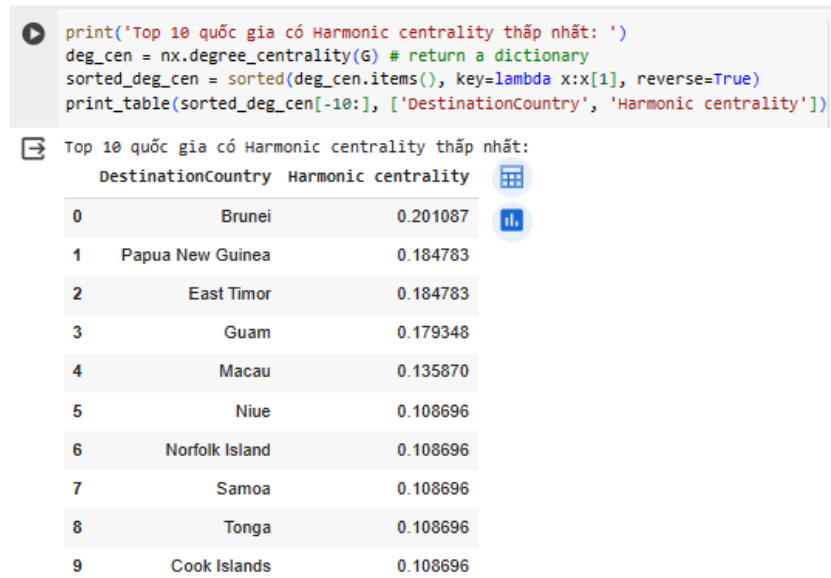
Hình 3: Biểu đồ thể hiện sự phân bố Harmonic Centrality của các node

```
print('Top 10 quốc gia có Harmonic centrality cao nhất: ')
deg_cen = nx.degree_centrality(G) # return a dictionary
sorted_deg_cen = sorted(deg_cen.items(), key=lambda x:x[1], reverse=True)
print_table(sorted_deg_cen[:10], ['DestinationCountry', 'Harmonic centrality'])
```

Top 10 quốc gia có Harmonic centrality cao nhất:

	DestinationCountry	Harmonic centrality
0	United States	1.000000
1	United Arab Emirates	1.000000
2	United Kingdom	1.000000
3	Germany	1.000000
4	Hong Kong	1.000000
5	Singapore	1.000000
6	Australia	0.994565
7	China	0.983696
8	Japan	0.983696
9	Thailand	0.972826

Hình 4: Liệt kê các quốc gia có giá trị Harmonic cao nhất



Hình 5: Liệt kê các quốc gia có giá trị Harmonic thấp nhất

3.2 Sử dụng Gephi

Id	Harmonic Closeness ...
United Stat...	1.0
United Kin...	1.0
Singapore	1.0
Germany	1.0
Hong Kong	1.0
United Ara...	1.0
Australia	0.997283
China	0.991848
Japan	0.991848
Thailand	0.986413
Switzerland	0.986413
Canada	0.983696
South Africa	0.980978
Kenya	0.978261

Hình 6: Top 10 quốc gia có Harmonic centrality cao nhất

Id	Harmonic Closeness ... ^
Norfolk Isl...	0.554348
Niue	0.554348
Samoa	0.554348
Cook Islands	0.554348
Tonga	0.554348
Macau	0.567935
Guam	0.589674
Papua Ne...	0.592391
East Timor	0.592391
Brunei	0.600543
Botswana	0.608696
Swaziland	0.608696
Lesotho	0.608696
New Caled...	0.611413
Vanuatu	0.611413
Fiji	0.649457
Burma	0.711957

Hình 7: Top 10 quốc gia có Harmonic centrality thấp nhất

4. Betweenness Centrality

Betweenness centrality được định nghĩa như tổng tỷ số của các đường đi ngắn nhất từ một node tới một node khác đi qua một node cho trước. Độ đo này dùng để xem xét khả năng chi phối các quan hệ giữa các nút khác trong mạng. Node có giá trị này càng lớn thì node đó sẽ có sự ảnh hưởng càng lớn đến việc phân bố cấu trúc các cụm hay nhóm trong mạng càng lớn.

Ý nghĩa: khi một người chọn hãng hàng không để bay ra nước ngoài, họ sẽ biết tới những hãng hàng không khác qua các nước có betweenness cao.

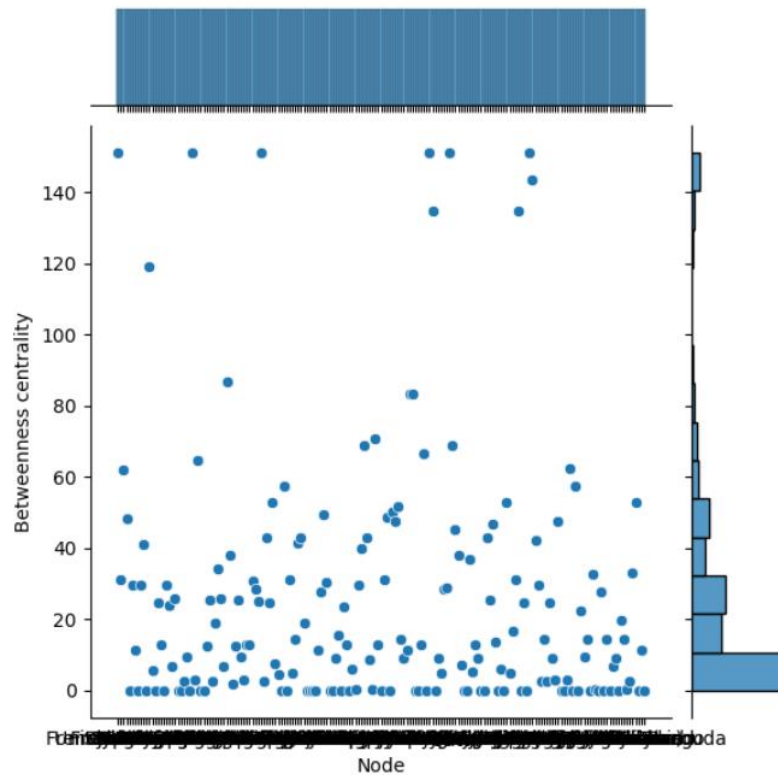
4.1 Sử dụng Python

```
def print_table(data, columns):
    df = pd.DataFrame(data, columns=columns)
    return df
bet_cen = nx.betweenness_centrality(G, normalized=False) #returns a dictionary
sorted_bet_cen = sorted(bet_cen.items(), key=lambda x:x[1],
reverse=True)
bet_cen_df = print_table(bet_cen.items(), ['Node', 'Betweenness
centrality'])
bet_cen_df
```

	Node	Betweenness centrality
0	United States	151.005506
1	Cyprus	31.095465
2	French Polynesia	61.812396
3	Jordan	48.411599
4	Swaziland	0.000000
...
180	New Zealand	33.170215
181	Oman	53.001829
182	Gibraltar	0.000000
183	Libya	11.267754
184	Aruba	0.000000

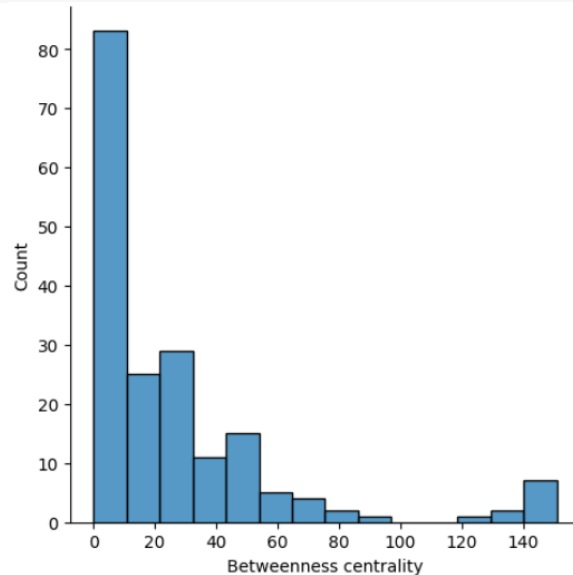
Hình 1: Kết quả tính giá trị Betweenness Centrality

```
#Biểu đồ thể hiện giá trị betweenness centrality
print(colored("Biểu đồ thể hiện giá trị betweenness centrality của
các node trong graph", 'green', attrs=['bold']))
sns.jointplot(x="Node", y="Betweenness centrality", data=bet_cen_df)
```



Hình 2: Biểu đồ thể hiện giá trị bậc của các node

```
#Biểu đồ thể hiện phân bố betweenness centrality của các node trong graph
print(colored("Biểu đồ thể hiện phân bố betweenness centrality của các node trong graph", 'green', attrs=['bold']))
sns.displot(x="Betweenness centrality", data=bet_cen_df)
```



Hình 3: Biểu đồ thể hiện sự phân bố Degree Centrality của các node

```
top_ten_max_bet_cen = sorted_bet_cen[:10]
print('Top 10 quốc gia có Betweenness centrality lớn nhất: ')
print_table(top_ten_max_bet_cen, ['DestinationCountry', 'Betweenness centrality'])
```

Top 10 quốc gia có Betweenness centrality lớn nhất:

	DestinationCountry	Betweenness centrality
0	United States	151.005506
1	United Arab Emirates	151.005506
2	United Kingdom	151.005506
3	Germany	151.005506
4	Hong Kong	151.005506
5	Singapore	151.005506
6	Australia	143.407282
7	China	134.620986
8	Japan	134.620986
9	Canada	118.983457

Hình 4: Liệt kê các quốc gia có giá trị Betweenness cao nhất

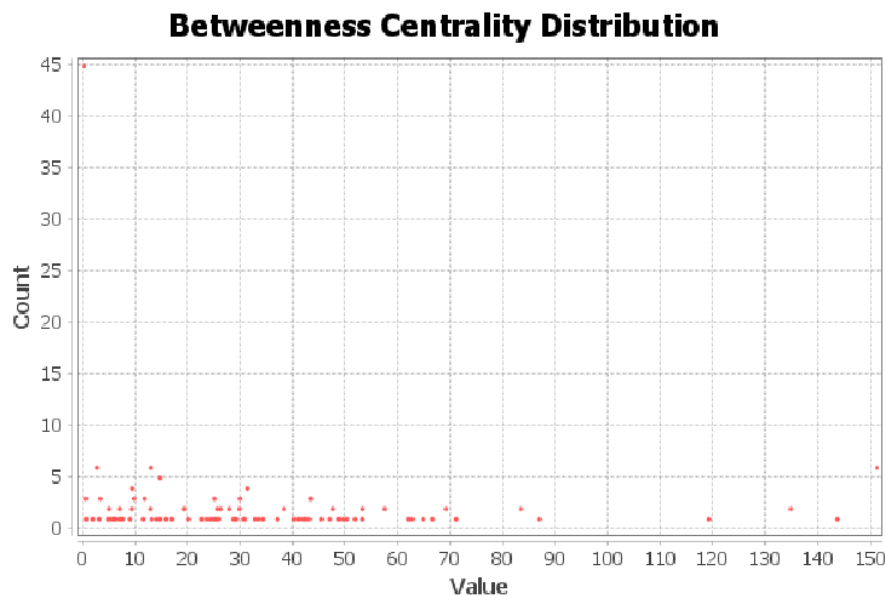

```
#Top 10 quốc gia có Betweenness centrality nhỏ nhất
top_ten_min_bet_cen = sorted_bet_cen[-10:]
print('Top 10 quốc gia có Betweenness centrality nhỏ nhất: ')
print_table(top_ten_min_bet_cen, ['DestinationCountry', 'Betweenness
centrality'])
```

Top 10 quốc gia có Betweenness centrality nhỏ nhất:

	DestinationCountry	Betweenness centrality
0	Cook Islands	0.0
1	Turkmenistan	0.0
2	Brunei	0.0
3	Guinea	0.0
4	Kyrgyzstan	0.0
5	Reunion	0.0
6	Trinidad and Tobago	0.0
7	East Timor	0.0
8	Gibraltar	0.0
9	Aruba	0.0

Hình 5: Liệt kê các quốc gia có giá trị Betweenness thấp nhất

4.2 Sử dụng Gephi



Hình 6: Biểu đồ phân phối theo giá trị Betweenness Centrality Distribution

Id	
United States	151.005506
United Arab Emirates	151.005506
United Kingdom	151.005506
Germany	151.005506
Hong Kong	151.005506
Singapore	151.005506
Australia	143.407282
China	134.620986
Japan	134.620986
Canada	118.983457

Hình 7: Top 10 quốc gia có giá trị Betweenness cao nhất

Cook Islands	0.0
Turkmenistan	0.0
Brunei	0.0
Guinea	0.0
Kyrgyzstan	0.0
Reunion	0.0
Trinidad and Tobago	0.0
East Timor	0.0
Gibraltar	0.0
Aruba	0.0

Hình 8: Top 10 quốc gia có giá trị Betweenness thấp nhất

5. Eigenvector Centrality

Eigenvector Centrality là thước đo mức độ ảnh hưởng của một nút trong mạng. Điểm tương đối được chỉ định cho tất cả các nút trong mạng dựa trên khái niệm rằng các kết nối đến các nút có điểm cao đóng góp nhiều hơn vào điểm số của nút được đề cập so với các kết nối ngang bằng với các nút có điểm thấp.

Ý nghĩa: các quốc gia có giá trị eigenvector cao được xem như một quốc gia trọng tâm trong mạng lưới, có mối quan hệ chặt chẽ với các quốc gia khác, là điểm đến của hầu hết các hãng hàng không.

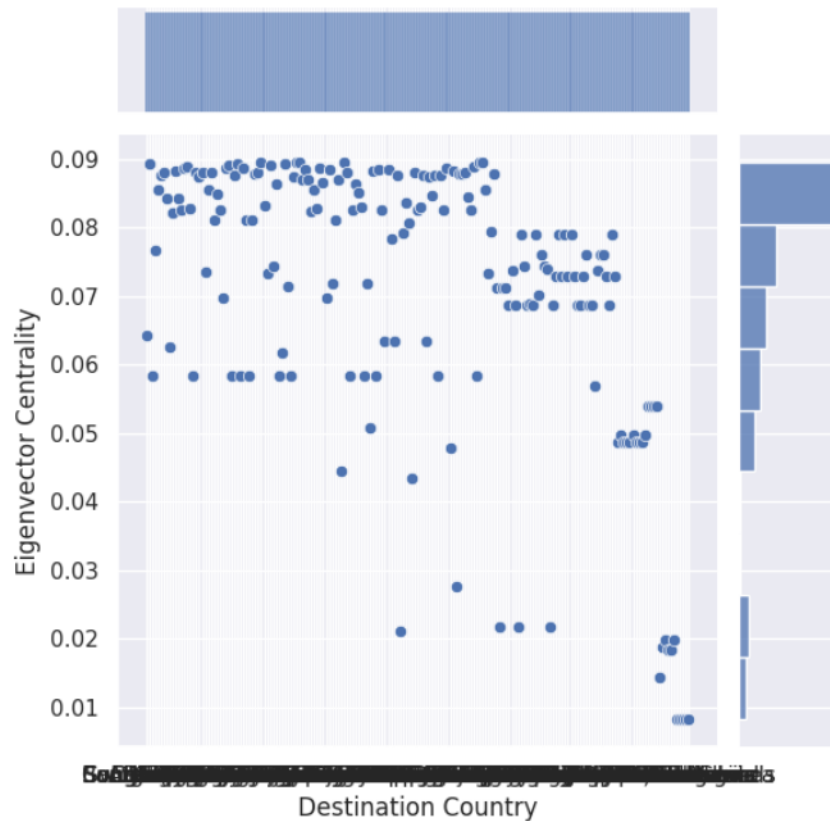
5.1 Sử dụng Python

```
eig_cen = nx.eigenvector_centrality(G, max_iter=100)
eig_cen_df = print_table(eig_cen.items(), ['Destination Country',
'Eigenvector Centrality'])
eig_cen_df
```

	Destination Country	Eigenvector Centrality
0	Bangladesh	0.0643092673
1	Switzerland	0.0892851397
2	South Sudan	0.0583541646
3	Ethiopia	0.0767084488
4	Malta	0.0856411851
...
180	Niue	0.0082369362
181	Samoa	0.0082369362
182	Norfolk Island	0.0082369362
183	Cook Islands	0.0082369362
184	Tonga	0.0082369362

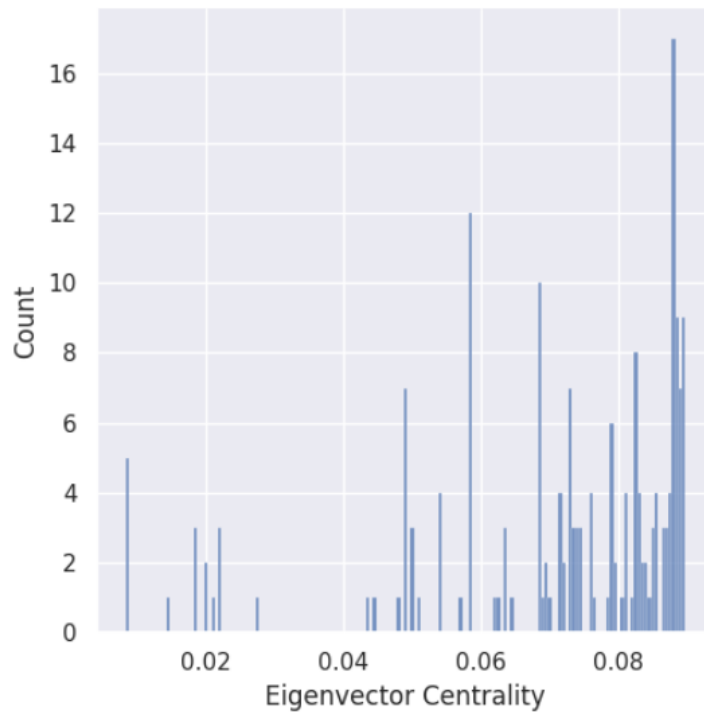
Hình 1: Kết quả tính giá trị Eigenvector Centrality

```
print(colored("Biểu đồ thể hiện giá trị Eigenvector centrality của  
các node trong graph", 'green', attrs=['bold']))  
sns.jointplot(x="Destination Country", y="Eigenvector Centrality",  
data=eig_cen_df)
```



Hình 2: Biểu đồ thể hiện giá trị bậc của các node

```
print(colored("Biểu đồ thể hiện phân bố Eigenvector Centrality của  
các node trong graph", 'green', attrs=['bold']))  
sns.displot(x="Eigenvector Centrality", data=eig_cen_df,  
binwidth=0.0005)
```



Hình 3: Biểu đồ thể hiện sự phân bố Eigenvector Centrality của các node

```
print('Top 6 quốc gia có Eigenvector Centrality lớn nhất: ')  
eig_cen_df = eig_cen_df.sort_values(by='Eigenvector Centrality',  
ascending=False)  
eig_cen_df.head(6)
```

Top 6 quốc gia có Eigenvector Centrality lớn nhất:

	Destination Country	Eigenvector Centrality
52	Singapore	0.0895678878
51	Hong Kong	0.0895678878
114	United Kingdom	0.0895678878
113	Germany	0.0895678878
67	United States	0.0895678878
39	United Arab Emirates	0.0895678878

Hình 4: Liệt kê các quốc gia có giá trị Eigenvector cao nhất

```
print('Top 5 quốc gia có Eigenvector Centrality nhỏ nhất: ')
eig_cen_df = eig_cen_df.sort_values(by='Eigenvector Centrality',
ascending=True)
eig_cen_df.head(5)
```

Top 5 quốc gia có Eigenvector Centrality nhỏ nhất:

	Destination Country	Eigenvector Centrality
184	Tonga	0.0082369362
180	Niue	0.0082369362
183	Cook Islands	0.0082369362
182	Norfolk Island	0.0082369362
181	Samoa	0.0082369362

Hình 5: Liệt kê các quốc gia có giá trị Eigenvector nhỏ nhất

5.2 Sử dụng Gephi

Id	Eigenvector Centrality ^
Tonga	0.092654
Niue	0.092654
Samoa	0.092654
Norfolk Island	0.092654
Cook Islands	0.092654

Hình 6: Các quốc gia có giá trị Eigenvector nhỏ nhất

Id	Eigenvector Centrality v
United Arab Emirates	1.0
Germany	1.0
United Kingdom	1.0
Hong Kong	1.0
Singapore	1.0
United States	1.0

Hình 7: Các quốc gia có giá trị Eigenvector lớn nhất

6. PageRank

PageRank là một thuật toán được phát triển bởi Larry Page và Sergey Brin, hai người sáng lập Google.

PageRank để đo lường và xác định tầm quan trọng của các node hoặc kết nối giữa các node trong mạng. Nó hoạt động bằng cách xem xét số lượng và chất lượng của các liên kết

đến node đó. Nếu một node có nhiều liên kết đến từ các node quan trọng khác, thì nó được coi là quan trọng hơn. Các node quan trọng hơn có khả năng xếp hạng cao hơn trong kết quả tìm kiếm.

Ý nghĩa: PageRank có độ đo càng cao thì các quốc gia đó càng có nhiều đường bay tới quốc gia khác trên một hãng hàng không và ngược lại.

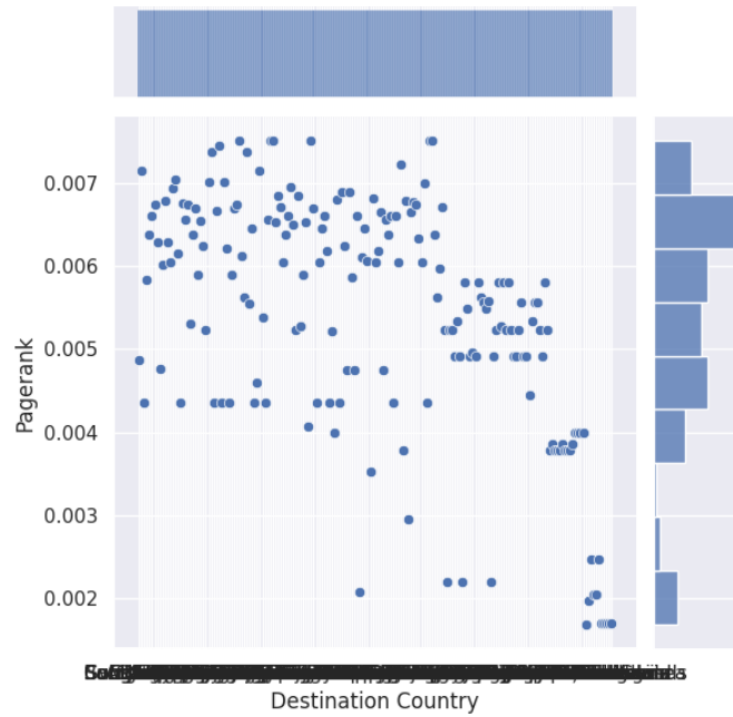
6.1 Sử dụng Python

```
pagerank = nx.pagerank(G, tol=1e-6, alpha=0.85)
pagerank_df = print_table(pagerank.items(), ['Destination Country',
'Pagerank'])
pagerank_df
```

	Destination Country	Pagerank
0	Bangladesh	0.0048710391
1	Switzerland	0.0071438780
2	South Sudan	0.0043516300
3	Ethiopia	0.0058303464
4	Malta	0.0063771997
...
180	Niue	0.0017059252
181	Samoa	0.0017059252
182	Norfolk Island	0.0017059252
183	Cook Islands	0.0017059252
184	Tonga	0.0017059252

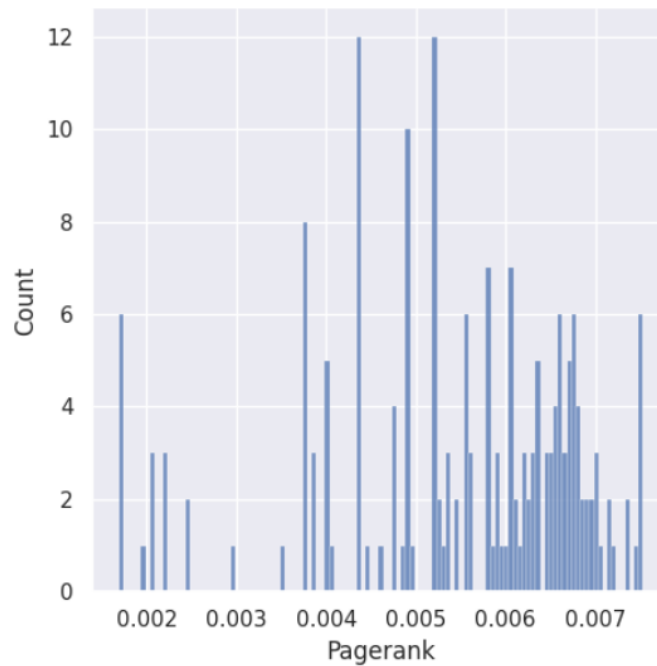
Hình 1: Kết quả tính giá trị PageRank

```
print(colored("Biểu đồ thể hiện giá trị pagerank của các node trong
graph", 'green', attrs=['bold']))
sns.jointplot(x="Destination Country", y="Pagerank",
data=pagerank_df)
```



Hình 2: Biểu đồ thể hiện giá trị bậc của các node

```
print(colored("Biểu đồ thể hiện phân bố pagerank của các node trong graph", 'green', attrs=['bold']))
sns.displot(x="Pagerank", data=pagerank_df, binwidth=0.00005)
```



Hình 3: Biểu đồ thể hiện sự phân bố của các node

```
print('Top 6 quốc gia có Pagerank lớn nhất \n')
pagerank_df = pagerank_df.sort_values(by='Pagerank',
ascending=False)
pagerank_df.head(6)
```

Top 6 quốc gia có Pagerank lớn nhất

	Destination Country	Pagerank
67	United States	0.0075059055
39	United Arab Emirates	0.0075059055
51	Hong Kong	0.0075059055
52	Singapore	0.0075059055
114	United Kingdom	0.0075059055
113	Germany	0.0075059055

Hình 4: Liệt kê các quốc gia có giá trị PageRank cao nhất

```
print('Quốc gia có Pagerank nhỏ nhất \n')
pagerank_df = pagerank_df.sort_values(by='Pagerank', ascending=True)
pagerank_df.head(1)
```

Quốc gia có Pagerank nhỏ nhất

	Destination Country	Pagerank
174	Macau	0.0016853291

Hình 5: Quốc gia có giá trị PageRank nhỏ nhất

6.2 Sử dụng Gephi

Id	PageRank ▾
Singapore	0.007511
United States	0.007511
Germany	0.007511
United Arab Emirates	0.007511
Hong Kong	0.007511
United Kingdom	0.007511

Hình 6: Các quốc gia có giá trị PageRank lớn nhất

Id	PageRank ▲
Macau	0.001686


Hình 7: Quốc gia có giá trị PageRank nhỏ nhất

PHẦN 3: CÁC THUẬT TOÁN GOM CỤM

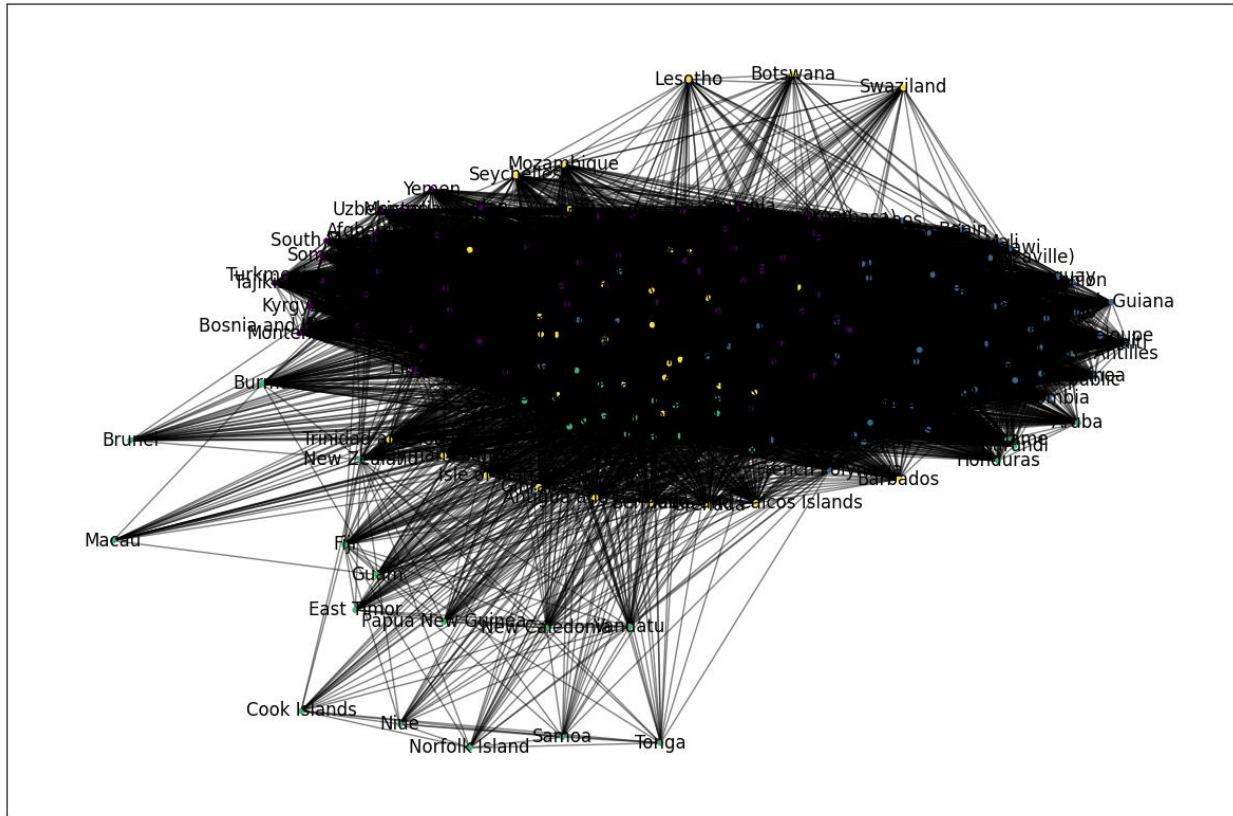
1. Thuật toán Louvain

Thuật toán Louvain là một thuật toán dùng để phân cụm trong các mạng phức tạp đặc biệt là trong mạng xã hội, mạng truyền thông hoặc mạng biểu đồ. Thuật toán Louvain dựa trên Modularity (một độ đo lường mức độ tập trung) để xác định cách tốt nhất để chia một mạng thành các cụm có tính liên kết mạnh bên trong và kết nối giữa các cụm. Thuật toán Louvain sử dụng phương pháp phân chia cộng đồng và thực hiện lặp đi lặp lại việc phân chia cộng đồng nhiều lần để có được mô đun tối đa của toàn bộ mạng. Thuật toán Louvain cần lặp lại liên tục để tìm ra cộng đồng tốt nhất cho mỗi đỉnh đến khi cộng đồng của tất cả các đỉnh không còn thay đổi.

Ưu điểm của thuật toán này là nhanh chóng và hiệu quả cho các mạng lớn.

 Sử dụng Python

```
plt.figure(figsize=(17,12))  
  
#compute the best partition  
partition = community_louvain.best_partition(G)  
  
#draw the graph  
pos = nx.spring_layout(G)  
  
#color the nodes according to their partition  
fig = plt.figure(figsize = (15,10), dpi=100)  
cmap = cm.get_cmap('viridis', max(partition.values())+1)  
  
nx.draw_networkx_nodes(G, pos, partition.keys(), node_size=10, cmap=cmap,  
node_color=list(partition.values()))  
  
nx.draw_networkx_edges(G, pos, alpha=0.5)  
  
nx.draw_networkx_labels(G, pos)  
  
plt.show()
```



Hình 1: Các quốc gia sau khi chia cụm

```
clusters = set([cluster for _, cluster in partition.items()])
print('Số cụm trong graph là: ', len(clusters))
```


```
Số cụm trong graph là: 4
```

Hình 2: Đếm số cụm có trong đồ thị

```
cluster_df = print_table(partition.items(), ['Node', 'Cluster'])
cluster_df
```

	Node	Cluster
0	Yemen	0
1	Portugal	1
2	Malaysia	0
3	Poland	2
4	Senegal	0
...
180	Botswana	3
181	Mozambique	3
182	Brunei	2
183	Papua New Guinea	2
184	East Timor	2

Hình 3: In ra bảng các node trong từng cụm trong đồ thị

 Bảng chi tiết các cụm

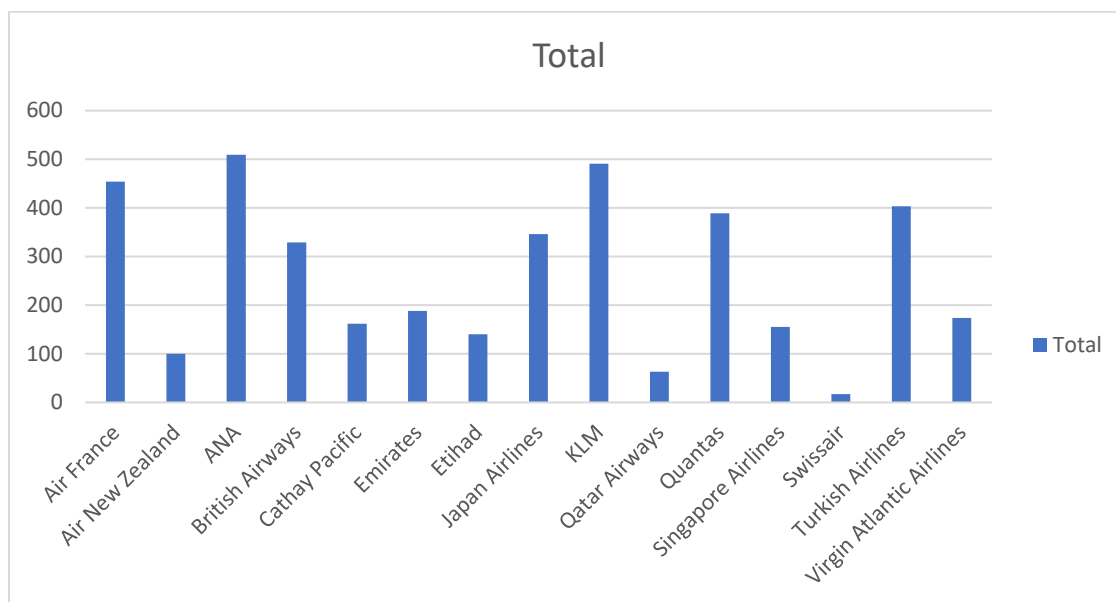
Cụm 1	Cụm 2	Cụm 3	Cụm 4
Afghanistan	Algeria	Poland	Zimbabwe
Albania	Angola	United Kingdom	Kenya
Argentina	Armenia	Switzerland	Russia
Azerbaijan	Austria	Thailand	Czech Republic
Bangladesh	Bahamas	United States	Uganda
Bosnia and Herzegovina	Belgium	United Arab Emirates	Oman
Bulgaria	Benin	Canada	South Africa
Burkina Faso	Cambodia	China	Kuwait
Cameroon	Central African Republic	Japan	India
Chad	Chile	Indonesia	Ireland
Congo (Kinshasa)	Colombia	Australia	Sweden
Cote d'Ivoire	Comoros	Turkey	Denmark
Croatia	Congo (Brazzaville)	Germany	Brazil
Djibouti	Costa Rica	Hong Kong	Italy
Equatorial Guinea	Cuba	South Korea	France
Estonia	Cyprus	Singapore	Saudi Arabia
Ethiopia	Dominican Republic	Mexico	Egypt
Finland	Ecuador	Burma	Nigeria
Gabon	French Guiana	Suriname	Qatar
Georgia	French Polynesia	Aruba	Bermuda

Ghana	Greece	Macau	Barbados
Hungary	Guadeloupe	Honduras	Saint Kitts and Nevis
Iran	Guatemala	New Zealand	Isle of Man
Iraq	Guinea	Guam	Turks and Caicos Islands
Israel	Haiti	Burundi	Namibia
Kazakhstan	Jamaica	Norfolk Island	Trinidad and Tobago
Kyrgyzstan	Jersey	Samoa	Seychelles
Latvia	Jordan	Tonga	Cayman Islands
Libya	Laos	Fiji	Grenada
Lithuania	Liberia	Vanuatu	Gibraltar
Luxembourg	Madagascar	Niue	Antigua and Barbuda
Macedonia	Malawi	New Caledonia	Lesotho
Malaysia	Mali	Cook Islands	Swaziland
Maldives	Martinique	Brunei	Botswana
Malta	Mauritius	Papua New Guinea	Mozambique
Mauritania	Mayotte	East Timor	
Moldova	Netherlands		
Mongolia	Netherlands Antilles		
Montenegro	Palau		
Morocco	Panama		
Nepal	Peru		
Niger	Philippines		
Norway	Portugal		
Pakistan	Puerto Rico		
Romania	Reunion		
Rwanda	Saint Lucia		
Senegal	Sierra Leone		
Serbia	Spain		
Slovenia	Taiwan		
Somalia	Togo		
South Sudan	Uruguay		
Sri Lanka	Virgin Islands		
Sudan	Zambia		
Tajikistan			
Tanzania			
Tunisia			
Turkmenistan			
Ukraine			
Uzbekistan			
Vietnam			

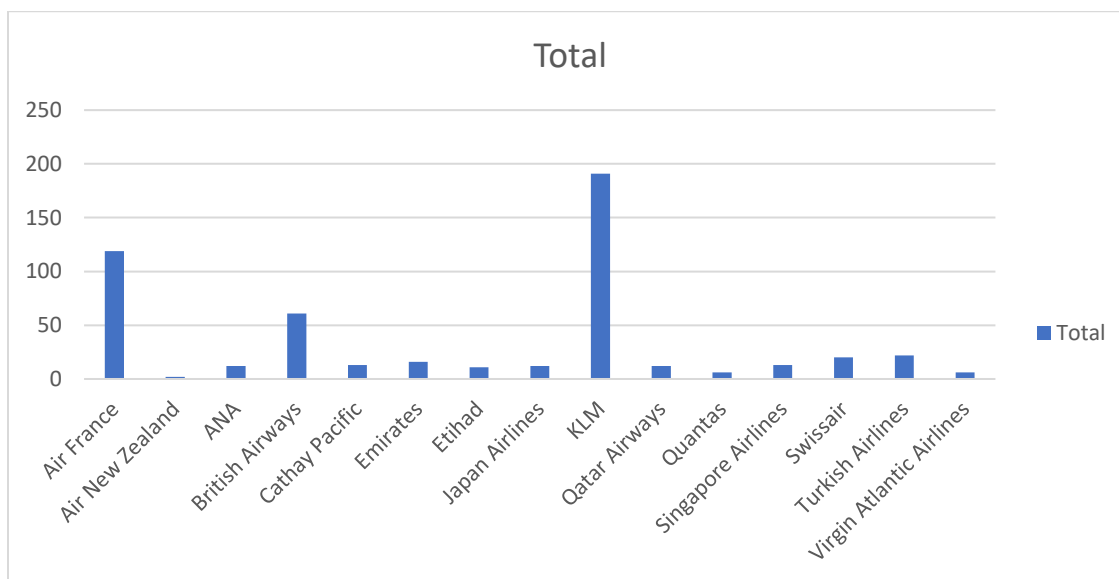
Yemen			
-------	--	--	--

📊 Sử dụng Excel analyze trực quan hóa các cụm

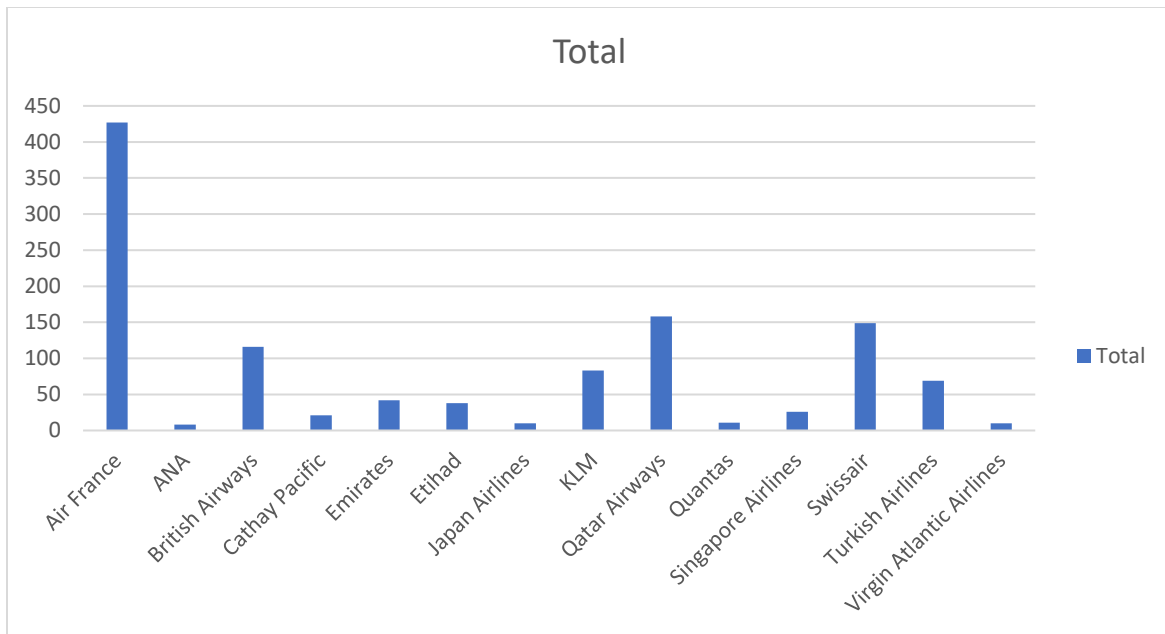
Cụm 1



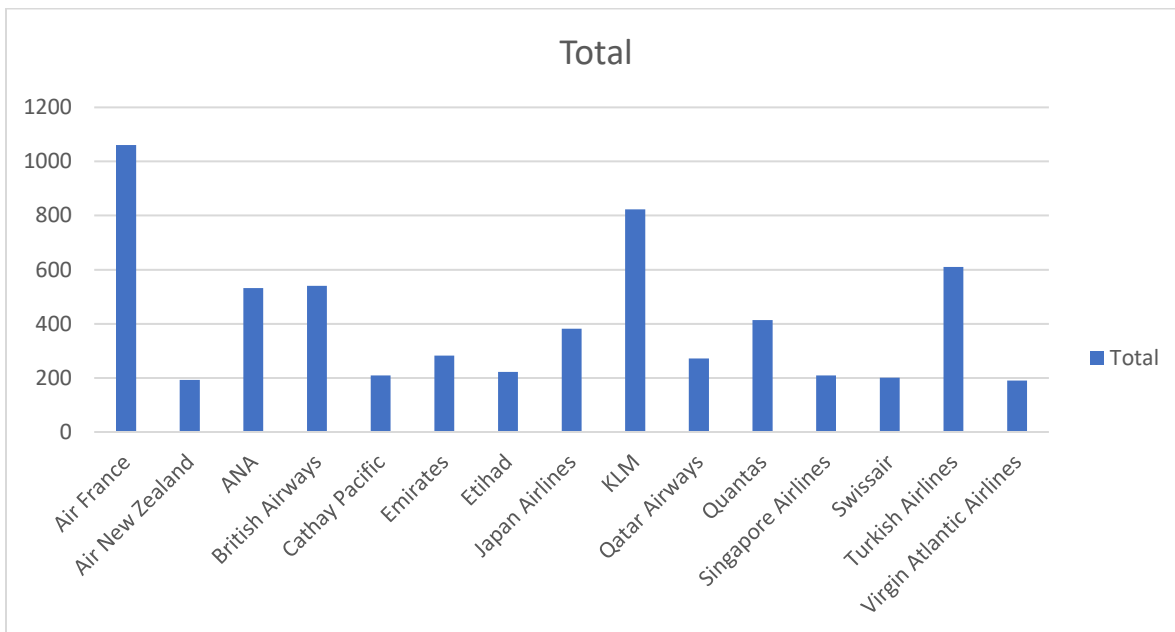
Cụm 2:



Cụm 3:



Cụm 4:



2. Thuật toán Girvan Newman

Thuật toán Girvan Newman là một thuật toán được sử dụng để phân tách hoặc phát hiện cộng đồng trong các mạng phức tạp. Girvan Newman dựa vào khái niệm Bottleneck nghĩa là các cạnh quan trọng mà bị loại bỏ sẽ ảnh hưởng lớn đến việc kết nối giữa các cộng đồng; bằng cách lặp lại quá trình loại bỏ cạnh quan trọng có thể tách mạng thành các cộng đồng con. Một đường đi giữa 2 đỉnh trong mạng thuộc 2 cộng đồng khác nhau nhất thiết phải đi qua ít nhất một trong số các cạnh như vậy, vì thế nếu ta thiết lập các đường đi giữa tất cả

các đỉnh trong mạng và xác định được cạnh nào trong đồ thị được đi qua nhiều nhất, ta có thể loại bỏ nó để phân chia mạng thành cộng đồng riêng biệt.

Ưu điểm của thuật toán này là có thể tìm ra cấu trúc cộng đồng trong mạng mà không cần thiết biết trước về số lượng cộng đồng hay cấu trúc dự kiến. Tuy nhiên thuật toán cũng có nhược điểm: do không biết trước về số lượng cộng đồng và với rất nhiều phép phân vùng thì khó có thể xác định được phép phân vùng nào mang lại hiệu quả tốt nhất; thuật toán có độ phức tạp thời gian khá cao với tổng cộng m cạnh cần loại bỏ cùng với thời gian tính toán là $O(mn)$ trong mỗi bước lặp do đó tổng thời gian tính toán có thể là $O(m^2n)$; thuật toán cũng không giải quyết được sự chồng chéo cộng đồng.

Sử dụng Python

```
#Định nghĩa 'edge_to_remove' và xác định cạnh có độ trung tâm
betweenness centrality cao nhất trong đồ thị
def edge_to_remove(graph):
    G_dict = nx.edge_betweenness centrality(graph)
    edge = ()
    # extract the edge with highest edge betweenness centrality score
    for key, value in sorted(G_dict.items(), key=lambda item: item[1],
reverse = True):
        edge = key
        break
    return edge
#Triển khai thuật toán Girvan-Newman để phân tách đồ thị thành các
cụm con
def girvan_newman(graph):
    # find number of connected components
    sg = nx.connected_components(graph)
    sg_count = nx.number_connected_components(graph)
    while(sg_count == 1):
        graph.remove_edge(edge_to_remove(graph)[0],
edge_to_remove(graph)[1])
        sg = nx.connected_components(graph)
        sg_count = nx.number_connected_components(graph)
    return sg
# tìm các cụm trong đồ thị G
c = girvan_newman(G.copy())

node_groups = []

for i in c:
    node_groups.append(list(i))
```

```

#Vẽ đồ thị với các cụm (communities) đã được phân loại trước đó
color_map = []
for node in G:
    if node in node_groups[0]:
        color_map.append('blue')
    else:
        color_map.append('red')

pd.option_context('precision',15)

nodes = G.nodes()
degree = G.degree()
colors = [degree[n] for n in nodes]

pos = nx.spring_layout(G)
fig = plt.figure(figsize = (10,10), dpi=100)

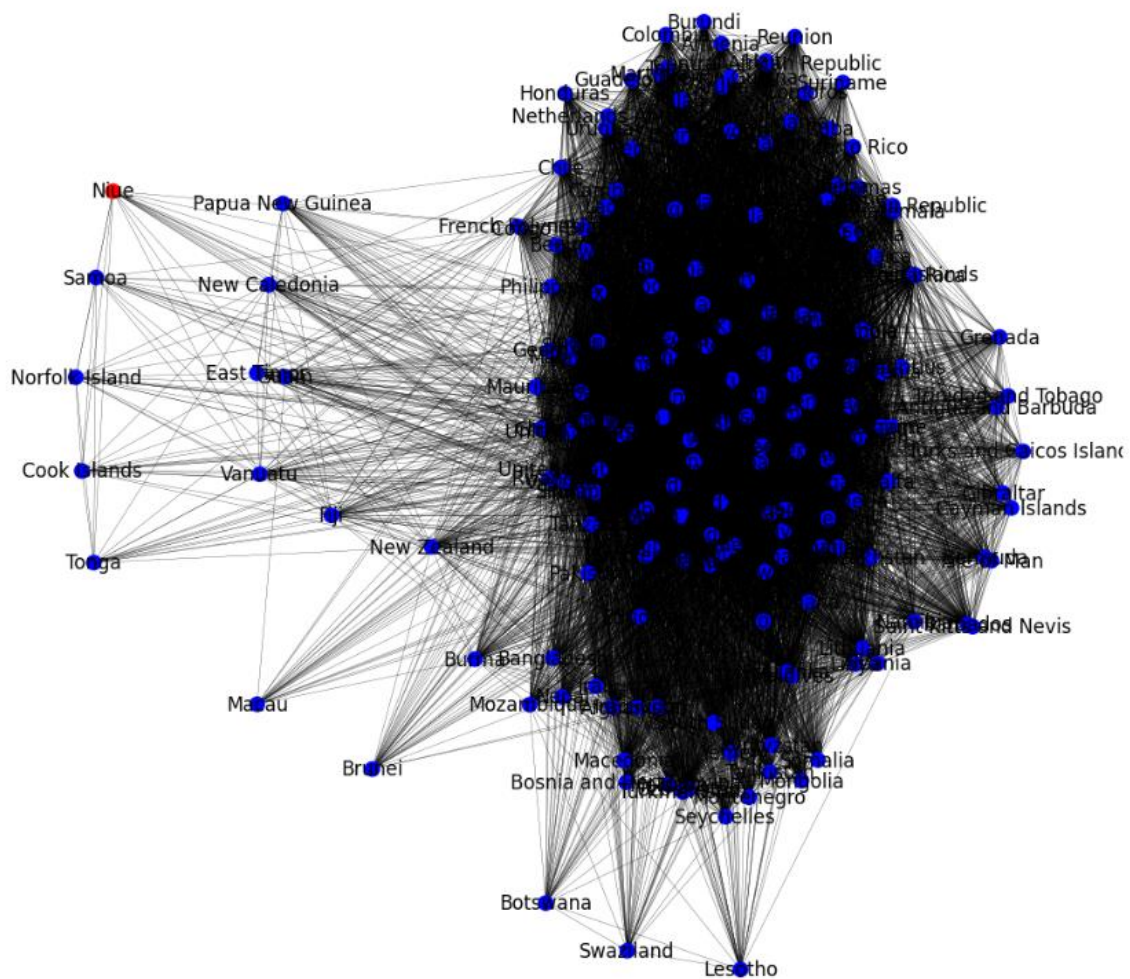
cmap = plt.cm.viridis_r
cmap = plt.cm.Greys

vmin = min(colors)
vmax = max(colors)

nx.draw(G, node_color=color_map, with_labels=True,node_size =
80,width = 0.2)

plt.legend()
plt.show()

```

Hình 1: Đồ thị phân cụm bằng Girvan Newman

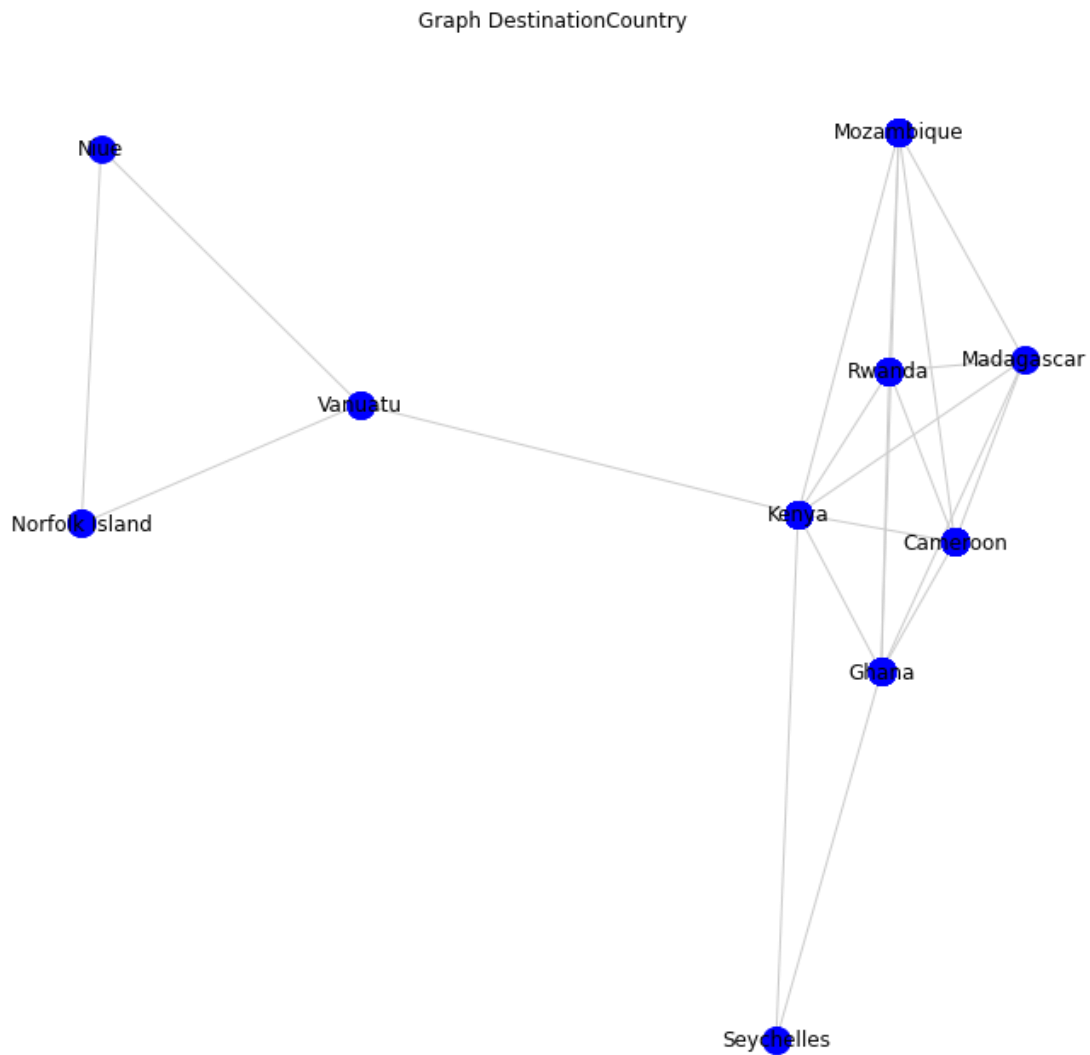
Dựa vào đồ thị, ta có:

- Cụm 1 (quốc gia có node màu đỏ): Niue
- Cụm 2 (quốc gia có node màu xanh): các quốc gia còn lại

⇒ Điều này thể hiện các chuyến bay rất khó khăn để di chuyển tới quốc gia Niue. Ngược lại, các chuyến bay di chuyển giữa các quốc gia còn lại dễ dàng hơn.

PHẦN 4: TÍNH TAY CÁC THUẬT TOÁN TRÊN 10 NODE NGẪU NHIÊN

Đồ thị biểu diễn liên kết giữa 10 node ngẫu nhiên:



1. Degree Centrality

nodes	số cạnh kề	nodes	Degree Centrality
Norfolk Island	2	Norfolk Island	0.22222222
Niue	2	Niue	0.22222222
Vanuatu	3	Vanuatu	0.33333333
Kenya	7	Kenya	0.77777778
Seychelles	2	Seychelles	0.22222222
Ghana	6	Ghana	0.66666667
Cameroon	5	Cameroon	0.55555556
Mzambique	5	Mzambique	0.55555556
Madagascar	5	Madagascar	0.55555556
Rwanda	5	Rwanda	0.55555556

10 node và tổng số cạnh liền kề mỗi node *Kết quả*

2. Closeness Centrality

Ma Trận Kề										
	Kenya	Mozambique	Rwanda	Madagascar	Cameroon	Ghana	Seychelles	Vanuatu	Norfolk Isl	Niue
Kenya	0	1	1	1	1	1	1	1	1	2
Mozambique	1	0	1	1	1	1	1	2	2	3
Rwanda	1	2	0	1	1	1	1	2	2	3
Madagascar	1	1	1	0	1	1	2	2	2	3
Cameroon	1	1	1	1	0	1	2	2	2	3
Ghana	1	1	1	1	1	0	1	2	3	3
Seychelles	1	2	2	2	2	1	0	2	3	3
Vanuatu	1	2	2	2	2	2	2	0	1	1
Norfolk Island	2	3	3	3	3	3	3	1	0	1
Niue	2	3	3	3	3	3	3	1	1	0
Sum of Distances	Closeness Centrality									
11	0.818181818									
15	0.6									
16	0.5625									
15	0.6									
15	0.6									
14	0.642857143									
18	0.5									
15	0.6									
22	0.409090909									
22	0.409090909									

Kết quả

3. Harmonic Centrality

Iteration 1																				
	Norfolk Island	Niue	Vanuatu	Kenya	Seychelles	Ghana	Cameroon	Mzambique	Madagascar	Rwanda										
Norfolk Island	0	1	1	0	0	0	0	0	0	0		1		2						0.139346603
Niue	1	0	1	0	0	0	0	0	0	0		1		2						0.139346603
Vanuatu	1	1	0	1	0	0	0	0	0	0		1		3						0.209019904
Kenya	0	0	1	0	1	1	1	1	1	1		1		7						0.487713110
Seychelles	0	0	0	1	0	1	0	0	0	0	X	1	=	2	/	14.3527	=			0.139346603
Ghana	0	0	0	1	1	0	1	1	1	1		1		6						0.418039809
Cameroon	0	0	0	1	0	1	0	1	1	1		1		5						0.348366507
Mzambique	0	0	0	1	0	1	1	0	1	1		1		5						0.348366507
Madagascar	0	0	0	1	0	1	1	1	1	0		1		5						0.348366507
Rwanda	0	0	0	1	0	1	1	1	1	1		1		5						0.348366507
Iteration 2																				
	Norfolk Island	Niue	Vanuatu	Kenya	Seychelles	Ghana	Cameroon	Mzambique	Madagascar	Rwanda										
Norfolk Island	0	1	1	0	0	0	0	0	0	0		0.139346603		0.348366507						0.068927826
Niue	1	0	1	0	0	0	0	0	0	0		0.139346603		0.209019904						0.041356696
Vanuatu	1	1	0	1	0	0	0	0	0	0		0.209019904		0.766406316						0.151641218
Kenya	0	0	1	0	1	1	1	1	1	1		0.487713110		2.159872344						0.427352523
Seychelles	0	0	0	1	0	1	0	0	0	0	X	0.139346603	=	0.905752919	/	5.0540765	=			0.179212348
Ghana	0	0	0	1	1	0	1	1	1	1		0.418039809		2.020525741						0.399781392
Cameroon	0	0	0	1	0	1	0	1	1	1		0.348366507		1.950852440						0.385995827
Mzambique	0	0	0	1	0	1	1	0	1	1		0.348366507		1.950852440						0.385995827
Madagascar	0	0	0	1	0	1	1	1	0	1		0.348366507		1.950852440						0.385995827
Rwanda	0	0	0	1	0	1	1	1	1	0		0.348366507		1.950852440						0.385995827
Iteration 3																				
	Norfolk Island	Niue	Vanuatu	Kenya	Seychelles	Ghana	Cameroon	Mzambique	Madagascar	Rwanda										
Norfolk Island	0	1	1	0	0	0	0	0	0	0		0.068927826		0.192997913						0.037424193
Niue	1	0	1	0	0	0	0	0	0	0		0.041356696		0.151641218						0.029404723
Vanuatu	1	1	0	1	0	0	0	0	0	0		0.151641218		0.537637045						0.104253109
Kenya	0	0	1	0	1	1	1	1	1	1		0.427352523		2.274618266						0.441070848
Seychelles	0	0	0	1	0	1	0	0	0	0	X	0.179212348	=	0.827133915	/	5.1570361	=			0.160389399
Ghana	0	0	0	1	1	0	1	1	1	1		0.399781392		2.150548179						0.417012438
Cameroon	0	0	0	1	0	1	0	1	1	1		0.385995827		1.985121396						0.384934558
Mzambique	0	0	0	1	0	1	1	0	1	1		0.385995827		1.985121396						0.384934558
Madagascar	0	0	0	1	0	1	1	1	0	1		0.385995827		1.985121396						0.384934558
Rwanda	0	0	0	1	0	1	1	1	1	0		0.385995827		1.985121396						0.384934558
Iteration 4																				
	Norfolk Island	Niue	Vanuatu	Kenya	Seychelles	Ghana	Cameroon	Mzambique	Madagascar	Rwanda										
Norfolk Island	0	1	1	0	0	0	0	0	0	0		0.037424193		0.133657833						0.025842779
Niue	1	0	1	0	0	0	0	0	0	0		0.029404723		0.104253109						0.020157367
Vanuatu	1	1	0	1	0	0	0	0	0	0		0.104253109		0.507899764						0.098202559
Kenya	0	0	1	0	1	1	1	1	1	1		0.441070848		2.221393179						0.429506980
Seychelles	0	0	0	1	0	1	0	0	0	0	X	0.160389399	=	0.858083286	/	5.1719606	=			0.165910639
Ghana	0	0	0	1	1	0	1	1	1	1		0.417012438		2.141198479						0.414001313
Cameroon	0	0	0	1	0	1	0	1	1	1		0.384934558		2.012886960						0.389192245
Mzambique	0	0	0	1	0	1	1	0	1	1		0.384934558		2.012886960						0.389192245
Madagascar	0	0	0	1	0	1	1	1	0	1		0.384934558		2.012886960						0.389192245
Rwanda	0	0	0	1	0	1	1	1	1	0		0.384934558		2.012886960						0.389192245
Iteration 5																				
	Norfolk Island	Niue	Vanuatu	Kenya	Seychelles	Ghana	Cameroon	Mzambique	Madagascar	Rwanda										
Norfolk Island	0	1	1	0	0	0	0	0	0	0		0.025842779		0.118359926						0.022877847
Niue	1	0	1	0	0	0	0	0	0	0		0.020157367		0.098202559						0.018981619
Vanuatu	1	1	0	1	0	0	0	0	0	0		0.098202559		0.475507126						0.091910999
Kenya	0	0	1	0	1	1	1	1	1	1		0.42950698		2.234883492						0.431981697
Seychelles	0	0	0	1	0	1	0	0	0	0	X	0.165910639	=	0.843508293	/	5.1735606	=			0.163042120
Ghana	0	0	0	1	1	0	1	1	1	1		0.414001313		2.152186601						0.415997175
Cameroon	0	0	0	1	0	1	0	1	1	1		0.389192245		2.011085029						0.388723585
Mzambique	0	0	0	1	0	1	1	0	1	1		0.389192245		2.011085029						0.388723585
Madagascar	0	0	0	1	0	1	1	1	0	1		0.389192245		2.011085029						0.388723585
Rwanda	0	0	0	1	0	1	1	1	1	0		0.389192245		2.011085029						0.388723585
Iteration 6																				
	Norfolk Island	Niue	Vanuatu	Kenya	Seychelles	Ghana	Cameroon	Mzambique	Madagascar	Rwanda										
Norfolk Island	0	1	1	0	0	0	0	0	0	0		0.022877847		0.110892619						0.021433552
Niue	1	0	1	0	0	0	0	0	0	0		0.018981619		0.091910999						0.017764746
Vanuatu	1	1	0	1	0	0	0	0	0	0		0.091910999		0.473841162						0.091584987
Kenya	0	0	1	0	1	1	1	1	1	1		0.431981697		2.225844634						0.430215796
Seychelles	0	0	0	1	0	1	0	0	0	0	X	0.163042120	=	0.847978871	/	5.1737864	=			0.163899088
Ghana	0	0	0	1	1	0	1	1	1	1		0.415997175		2.149918157						0.415540571
Cameroon	0	0	0	1	0	1	0	1	1	1		0.388723585		2.014149626						0.389298953
Mzambique	0	0	0	1	0	1	1	0	1	1		0.388723585		2.014149626						0.389298953
Madagascar	0	0	0	1	0	1	1	1	0	1		0.388723585		2.014149626						0.389298953
Rwanda	0	0	0	1	0	1	1	1	1	0		0.388723585		2.014149626						0.389298953

Ma trận kề										
	Norfolk Island	Niue	Vanuatu	Kenya	Seychelles	Ghana	Cameroon	Mzambique	Madagascar	Rwanda
Norfolk Island	0	1	1	0	0	0	0	0	0	0
Niue	1	0	1	0	0	0	0	0	0	0
Vanuatu	1	1	0	1	0	0	0	0	0	0
Kenya	0	0	1	0	1	1	1	1	1	1
Seychelles	0	0	0	1	0	1	0	0	0	0
Ghana	0	0	0	1	1	0	1	1	1	1
Cameroon	0	0	0	1	0	1	0	1	1	1
Mzambique	0	0	0	1	0	1	1	0	1	1
Madagascar	0	0	0	1	0	1	1	1	0	1
Rwanda	0	0	0	1	0	1	1	1	1	0

Tính PageRank										
Nodes	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6	Iteration 7	Iteration 8	Iteration 9	Iteration 10
Norfolk Island	0.1	0.22083333	0.31387798	0.38950681	0.45042737	0.49986014	0.54010917	0.57301606	0.60002496	0.62226987
Niue	0.1	0.22083333	0.31387798	0.38950681	0.45042737	0.49986014	0.54010917	0.57301606	0.60002496	0.62226987
Vanuatu	0.1	0.24714286	0.37450119	0.47607167	0.55915946	0.62706567	0.6828341	0.72879929	0.76679739	0.79831071
Kenya	0.1	0.303	0.48815024	0.6430008	0.77578451	0.88902454	0.98581826	1.06839554	1.13885465	1.19892406
Seychelles	0.1	0.17630952	0.22541726	0.26825049	0.30418093	0.33527621	0.3618973	0.38476226	0.40434043	0.42110037
Ghana	0.1	0.27264286	0.41629488	0.5371924	0.64287395	0.73372463	0.81215832	0.87957662	0.93748857	0.98714341
Cameroon	0.1	0.22730952	0.34134512	0.4423365	0.52977254	0.60546021	0.67068201	0.72681009	0.77501358	0.8163573
Mzambique	0.1	0.22730952	0.34134512	0.4423365	0.52977254	0.60546021	0.67068201	0.72681009	0.77501358	0.8163573
Madagascar	0.1	0.22730952	0.34134512	0.4423365	0.52977254	0.60546021	0.67068201	0.72681009	0.77501358	0.8163573
Rwanda	0.1	0.22730952	0.34134512	0.4423365	0.52977254	0.60546021	0.67068201	0.72681009	0.77501358	0.8163573

PageRank	
Norfolk Island	0.62226987
Niue	0.62226987
Vanuatu	0.79831071
Kenya	1.19892406
Seychelles	0.42110037
Ghana	0.98714341
Cameroon	0.8163573
Mzambique	0.8163573
Madagascar	0.8163573
Rwanda	0.8163573

Kết quả

7. Thuật toán gom cụm Louvain

Ma trận cạnh:

Ma trận										
	Madagascar	Mozambique	Rwanda	Kenya	Cameroon	Ghana	Seychelles	Vanuatu	Niue	Norfolk Island
Madagascar	0	1	1	1	1	1	0	0	0	0
Mozambique	1	0	1	1	1	1	0	0	0	0
Rwanda	1	1	0	1	1	1	0	0	0	0
Kenya	1	1	1	0	1	1	1	1	0	0
Cameroon	1	1	1	1	0	1	0	0	0	0
Ghana	1	1	1	1	1	0	1	0	0	0
Seychelles	0	0	0	1	0	1	0	0	0	0
Vanuatu	0	0	0	1	0	0	0	0	1	1
Niue	0	0	0	0	0	0	0	1	0	1
Norfolk Island	0	0	0	0	0	0	0	1	1	0

Hình 1: Ma trận thu được từ đồ thị

Phân cụm:

Cụm 1	Kenya	7			Madagascar	Mozambique	Rwanda	Kenya	Cameroon	Ghana	Seychelles	Vanuatu	Niue	Norfolk Island
Cụm 2	Ghana	6		Madagascar	0	0.40476	0.4047619	0.16667	0.40476	0.28571	0	0	0	0
Cụm 3	Cameroon	5		Mozambique	0.40476	0	0.4047619	0.16667	0.40476	0.28571	0	0	0	0
Cụm 4	Madagascar	5		Rwanda	0.40476	0.40476	0	0.16667	0.40476	0.28571	0	0	0	0
Cụm 5	Mozambique	5		Kenya	0.16667	0.16667	0.1666667	0	0.16667	0	0.66667	0.5	0	0
Cụm 6	Rwanda	5		Cameroon	0.40476	0.40476	0.4047619	0.16667	0	0.28571	0	0	0	0
Cụm 7	Vanuatu	3		Ghana	0.28571	0.28571	0.2857143	0	0.28571	0	0.71429	0	0	0
Cụm 8	Seychelles	2		Seychelles	0	0	0	0.66667	0	0.71429	0	0	0	0
Cụm 9	Niue	2		Vanuatu	0	0	0	0.5	0	0	0	0	0.85714	0.85714
Cụm 10	Norfolk Island	2		Niue	0	0	0	0	0	0	0	0.85714	0	0.90476
		42		Norfolk Isl	0	0	0	0	0	0	0	0.85714	0.90476	0

Hình 2: Thực hiện chia mỗi đỉnh thành 1 cụm sau đó tính toán để phân chia ra cụm cuối cùng

Kết quả thu được:

Cụm 1	Mozambique
	Rwanda
	Cameroon
	Madagascar
	Gahana
	Kenya
	Seychelles
Cụm 2	Vanuatu
	Niue
	Norfolk Island

Hình 3: Kết quả

PHẦN 5: NGÔN NGỮ VÀ ỨNG DỤNG SẼ ÁP DỤNG

1. Ngôn ngữ

Trong đề tài lần này, nhóm em sẽ sử dụng ngôn ngữ lập trình Python, chạy trên hai nền tảng là Google Colab và Jupyter Notebook, nhằm mục đích tiến hành phân tích và cải thiện quá trình di chuyển trên hệ thống mạng hàng không “Air Travel Network”.

Sử dụng ngôn ngữ Python để phân tích mạng xã hội “Air Travel Network” mang lại một loạt lợi ích quan trọng bởi: Python phổ biến và dễ tích hợp, có sự linh hoạt cho việc tùy chỉnh, hỗ trợ từ cộng đồng mạnh mẽ, sử dụng các thư viện mạnh mẽ, và tích hợp tốt với các ứng dụng khác.

2. Ứng dụng

Bên cạnh việc sử dụng **Jupyter Notebook** và **Google Colab** để đáp ứng nhu cầu phân tích “Air Travel Network”, nhóm của chúng em đã tích hợp thêm ứng dụng **Gephi** và **Excel analyze** để trực quan hóa dữ liệu một cách rõ ràng hơn góp phần trong việc phân tích dữ liệu mạng.

2.1 Gephi

Gephi là một phần mềm mã nguồn mở được sử dụng để thực hiện phân tích mạng và hiển thị dữ liệu mạng. Ứng dụng Gephi chủ yếu dành cho người làm nghiên cứu mạng xã hội, phân tích mạng lưới, và các ngành công nghiệp khác mà yêu cầu phân tích và trực quan hóa dữ liệu mạng.

Gephi cung cấp các tính năng chính sau:

- *Trực quan hóa dữ liệu mạng*: Gephi cho phép nhập dữ liệu mạng từ nhiều nguồn khác nhau, như tệp CSV hoặc Excel, và sau đó biểu diễn dữ liệu mạng dưới dạng đồ thị, nơi các nút (nodes) biểu thị các thực thể (như người dùng hoặc trang web) và các cạnh (edges) biểu thị mối quan hệ giữa chúng.
- *Phân tích mạng*: Gephi cung cấp nhiều công cụ để thực hiện phân tích mạng, bao gồm tính toán các thước đo mạng như bậc tương tác, trung tâm tương tác, hay cụm mạng. Điều này giúp người dùng hiểu rõ hơn về cấu trúc và đặc điểm của mạng.
- *Tùy chỉnh trực quan hóa*: Gephi cho phép tùy chỉnh trực quan hóa của đồ thị, bao gồm màu sắc, kích thước, và vị trí của các nút để làm cho dữ liệu mạng dễ hiểu hơn.

- *Xuất dữ liệu:* Có thể xuất dữ liệu mạng đã trực quan hóa sang nhiều định dạng khác nhau để chia sẻ kết quả của mình với người khác hoặc sử dụng cho các mục đích khác .

Gephi thường được sử dụng trong nghiên cứu xã hội học, tiếp thị, thống kê, và các lĩnh vực khác đòi hỏi phân tích dữ liệu mạng.

2.2 Excel analyze

Excel analyze là phần mềm đã được tích hợp sẵn trong Excel, dùng để phân tích và trực quan hóa dữ liệu thông dụng, dễ sử dụng. Nhiệm vụ chính là tối giản hóa và nâng cao chất lượng việc thể hiện dữ liệu trên các báo cáo tài chính, doanh số, Hiện tại ứng dụng có mặt trên cả 2 nền tảng là website, máy tính; được tích hợp sẵn ở tất cả các phiên bản Excel, nhờ đó ứng dụng có thể đảm bảo đáp ứng nhu cầu mọi nơi mọi lúc cho người sử dụng.

TÀI LIỆU THAM KHẢO

Borgatti, S. P. (2006). *Identifying sets of key players in a social network*.

Developers, N. (2004-2023). *Centrality*. Retrieved from NetworkX Analysis in Python:
<https://networkx.org/documentation/stable/reference/algorithms/centrality.html>

Developers, N. (2004-2023). *Community*. Retrieved from NetworkX Analysis in Python:
<https://networkx.org/documentation/stable/reference/algorithms/community.html>

Developers, N. (2004-2023). *Community Detection using Girvan-Newman*. Retrieved from NetworkX Analysis in Python:
https://networkx.org/documentation/stable/auto_examples/algorithms/plot_girvan_newman.html

Developers, N. (2004-2023). *louvain_communities*. Retrieved from NetworkX Analysis in Python:
https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.louvain.louvain_communities.html

ĐHQG, Đ. h. (2023). *Chương 4: Độ đo trung tâm (Centrality measures) & tầm ảnh hưởng (Key players)*. Hồ Chí Minh.

ĐHQG, Đ. h. (2023). *Chương 5: Khám phá cộng đồng (Community Detection)*. Hồ Chí Minh.

Faust, S. W. (2009). *Social Network Analytics: Methods and Applications*.

Jackson, M. O. (n.d.). *Social and Economic Networks: Models and Analysis*.

LINK VIDEO TRÌNH BÀY VÀ DEMO

<https://drive.google.com/drive/folders/10RDP0spteLrTcLF4iXZAABsMW9a-pQoF?usp=sharing>

Bảng phân công công việc

	Trần Minh Ngọc	Nguyễn Anh Nguyên	Lê Thị Đoan Trang	Nguyễn Thị Thảo Trang
	STT 31	STT 33	STT 46	STT 47
Tìm hiểu đề tài	X	X	X	X
Phân chia công việc			X	
Thu thập và mô tả dữ liệu				X
Xử lý và phân tích dữ liệu	X			
Chuyển đổi dữ liệu		X		
Độ đo Degree Centrality (Python, Gephi, Tính tay)			X	
Độ đo Closeness Centrality (Python, Gephi, Tính tay)				X
Độ đo Harmonic Centrality (Python, Gephi, Tính tay)	X			
Độ đo Betweenness Centrality (Python, Gephi, Tính tay)		X		
Độ đo Eigenvector Centrality (Python, Gephi, Tính tay)			X	

PageRank (Python, Gephi, Tính tay)			X	
Thuật toán gom cụm Louvain (Python, Tính tay)	X			X
Thuật toán gom cụm Girvan Newman (Python, Tính tay)		X		
Làm doc báo cáo		X	X	
Làm slide báo cáo	X			X
Quay video demo	X	X	X	X