**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](#_Toc154942979)

[1. Giới thiệu đề tài 1](#_Toc154942980)

[2. Giới thiệu dữ liệu 1](#_Toc154942981)

[a. Nguồn dữ liệu 1](#_Toc154942982)

[b. Mô tả dữ liệu 1](#_Toc154942983)

[c. Xử lý và phân tích dữ liệu 3](#_Toc154942984)

[d. Chuyển đổi data thành đồ thị 5](#_Toc154942985)

[**PHẦN 2: CÁC ĐỘ ĐO TRUNG TÂM** 8](#_Toc154942986)

[1. Degree Centrality 8](#_Toc154942987)

[1.1 Sử dụng Python 8](#_Toc154942988)

[1.2 Sử dụng Gephi 11](#_Toc154942989)

[2. Closeness Centrality 11](#_Toc154942990)

[2.1 Sử dụng Python 11](#_Toc154942991)

[2.2 Sử dụng Gephi 14](#_Toc154942992)

[3. Harmonic Centrality 15](#_Toc154942993)

[3.1 Sử dụng Python 15](#_Toc154942994)

[3.2 Sử dụng Gephi 17](#_Toc154942995)

[4. Betweenness Centrality 18](#_Toc154942996)

[4.1 Sử dụng Python 18](#_Toc154942997)

[4.2 Sử dụng Gephi 22](#_Toc154942998)

[5. Eigenvector Centrality 23](#_Toc154942999)

[5.1 Sử dụng Python 23](#_Toc154943000)

[5.2 Sử dụng Gephi 26](#_Toc154943001)

[6. PageRank 26](#_Toc154943002)

[6.1 Sử dụng Python 27](#_Toc154943003)

[6.2 Sử dụng Gephi 29](#_Toc154943004)

[**PHẦN 3: CÁC THUẬT TOÁN GOM CỤM** 30](#_Toc154943005)

[1. Thuật toán Louvain 30](#_Toc154943006)

[2. Thuật toán Girvan Newman 35](#_Toc154943007)

[**PHẦN 4: TÍNH TAY CÁC THUẬT TOÁN TRÊN 10 NODE NGẪU NHIÊN** 39](#_Toc154943008)

[1. Degree Centrality 40](#_Toc154943009)

[2. Closeness Centrality 40](#_Toc154943010)

[3. Harmonic Centrality 40](#_Toc154943011)

[4. Betweenness Centrality 41](#_Toc154943012)

[5. Eigenvector Centrality 41](#_Toc154943013)

[6. PageRank 43](#_Toc154943014)

[7. Thuật toán gom cụm Louvain 44](#_Toc154943015)

[**PHẦN 5: NGÔN NGỮ VÀ ỨNG DỤNG SẼ ÁP DỤNG** 46](#_Toc154943016)

[1. Ngôn ngữ 46](#_Toc154943017)

[2. Ứng dụng 46](#_Toc154943018)

[**Tài liệu tham khảo** 48](#_Toc154943019)

**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.

1. Giới thiệu dữ liệu
   1. 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-world-none-in-us-qatar-2022-9)

- 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/datasets/flight-routes)

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

* 1. 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 screenshot of a computer

Description automatically generated

*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 screenshot of a computer

Description automatically generated

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

- File Airlines.csv:

A screenshot of a computer

Description automatically generated

*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 screenshot of a computer

Description automatically generated

*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 screenshot of a table

Description automatically generated

*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*

* 1. Xử lý và phân tích dữ liệu
     1. Làm sạch dữ liệu

1. Import các thư viện cần thiết

A screenshot of a computer program

Description automatically generated

*Hình: Các thư viện được import*

b) Thực hiện làm sạch dữ liệu

A screen shot of a computer

Description automatically generated

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

- Kết quả thu được:

A screenshot of a computer

Description automatically generated

*Hình: Kết quả thu được sau khi làm sạch data*

* 1. Chuyển đổi data thành đồ thị

1. Đồ thị một phía

* Chú thích đồ thị:

+ Node: các quốc gia

+ Edge: hai quốc gia cùng là điểm đến của hãng hàng không sẽ hợp thành 1 cạnh

+ Weight: số quốc gia cùng là điểm đến của các hãng hàng không

🡪 Điều này nói lên rằng mỗi hãng hàng không vận chuyển khách tới nhiều quốc gia khác nhau, tại mỗi quốc gia có sự cạnh tranh giữa các hãng hàng không.

A computer screen with white text

Description automatically generated

*Hình: Code hiển thị đồ thị 1 phía*

* Kết quả:

A network of blue dots and lines

Description automatically generated

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

A computer screen shot of text

Description automatically generated

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

- Kết quả thu được:

A screenshot of a table

Description automatically generated

*Hình: File csv sau khi xuất*

1. 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:

A screenshot of a computer program

Description automatically generated

*Hình: Code hiển thị đồ thị 2 phía*

* Kết quả:

A graph of lines and dots

Description automatically generated with medium confidence

**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.

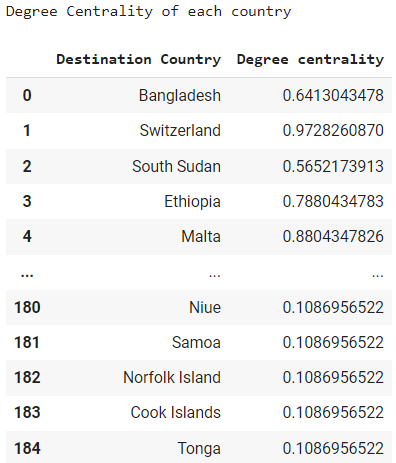
* 1. Sử dụng Python

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")

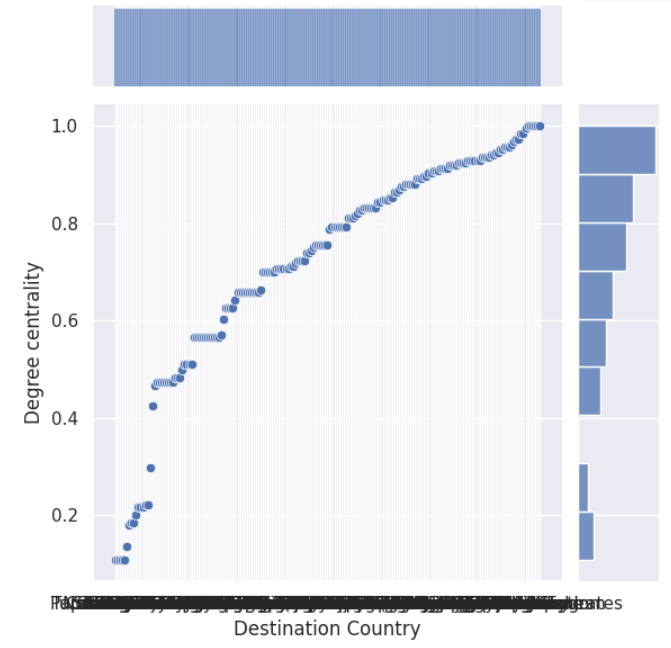
deg\_cen\_df



*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']))

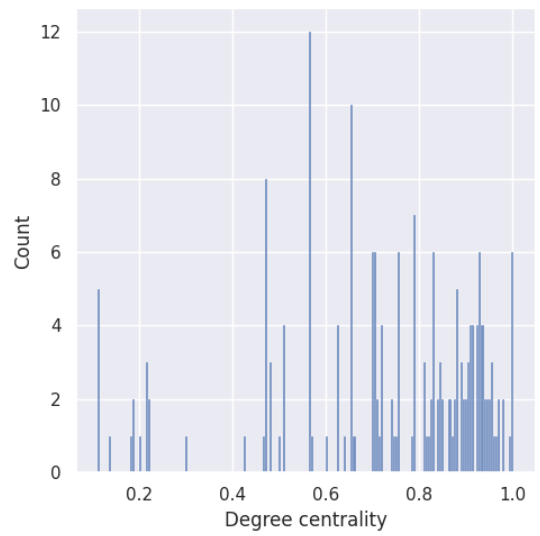
sns.jointplot(x="Destination Country", y="Degree centrality", data=deg\_cen\_df)



*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)

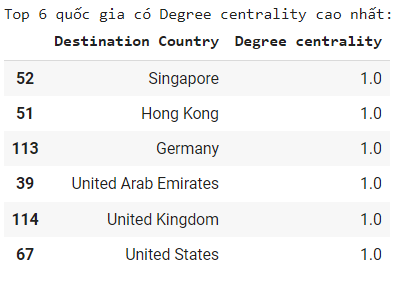


*Hình 3: Biểu đồ thể hiện sự phân bố Degree Centrality của các node*

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)

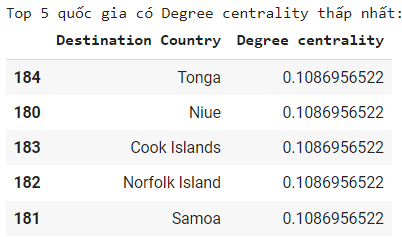


*Hình 4: Liệt kê các quốc gia có giá trị Degree cao nhất*

print('Top 5 quốc gia có Degree centrality thấp nhất: ')

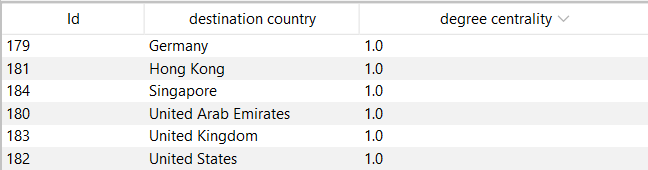
deg\_cen\_df = deg\_cen\_df.sort\_values(by='Degree centrality', ascending=True)

deg\_cen\_df.head(5)

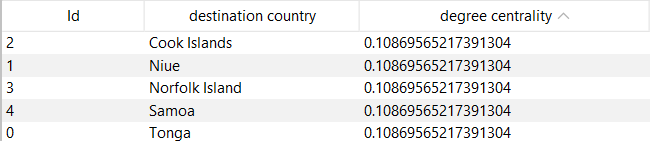


*Hình 5: Liệt kê các quốc gia có giá trị Degree thấp nhất*

* 1. Sử dụng Gephi



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

**

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

1. Closeness Centrality

Closeness Centralitylà 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.

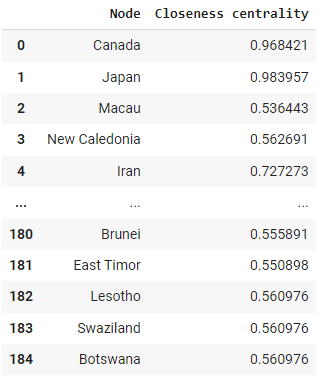
* 1. Sử dụng Python

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'])

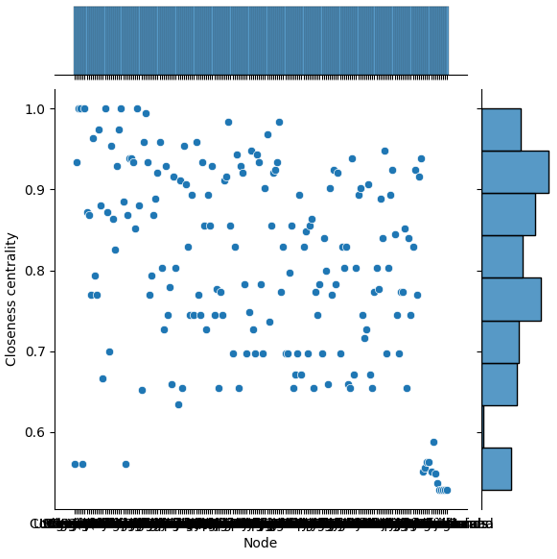
closeness\_centrality\_df



*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']))

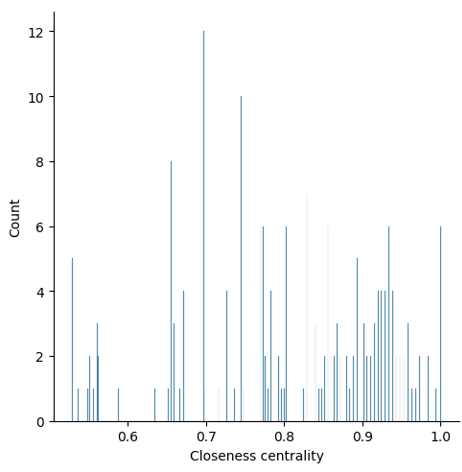
sns.jointplot(x="Node",y="Closenesscentrality", data=closeness\_centrality\_df)



*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)

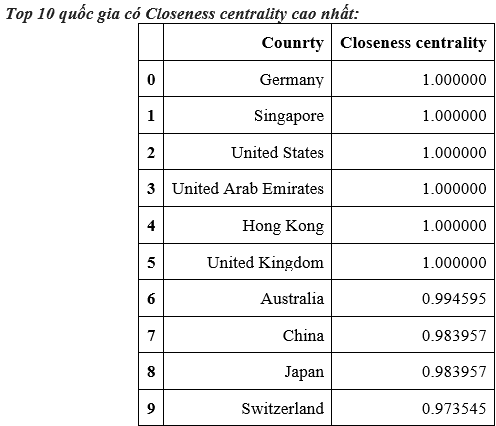
**

*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, ['Counrty', 'Closeness centrality'])

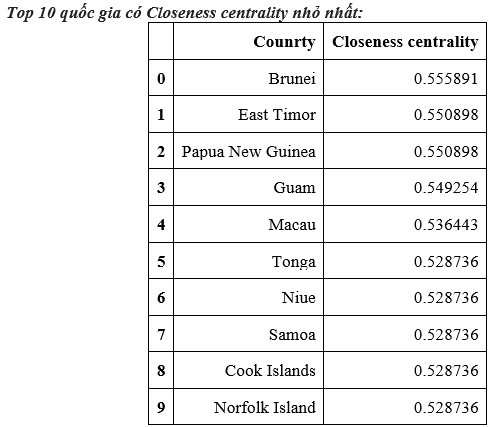
**

*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'])

**

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

* 1. Sử dụng Gephi

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

1. Harmonic Centrality

Harmonic Centralitycòn 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.

* 1. Sử dụng Python

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

*Hình 1: Kết quả tính giá trị Harmonic Centrality*

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer screen

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

* 1. Sử dụng Gephi

*A screenshot of a computer

Description automatically generated*

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

*A screenshot of a table

Description automatically generated*

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

1. 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.

* 1. Sử dụng Python

def print\_table(data, columns):

  df = pd.DataFrame(data, columns=columns)

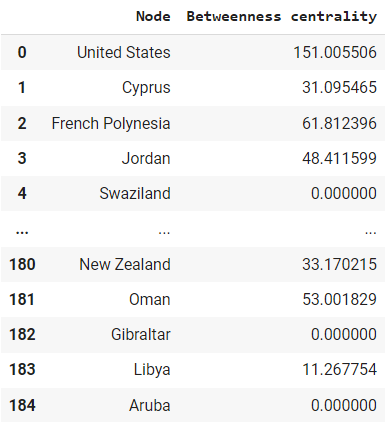
  return df

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

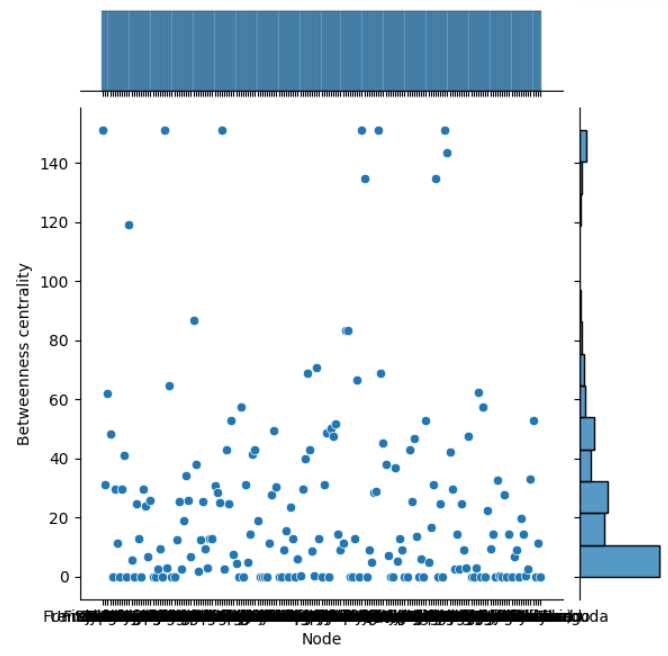


*Hình 1: Kết quả tính giá trị Betweenness Centrality*

#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)

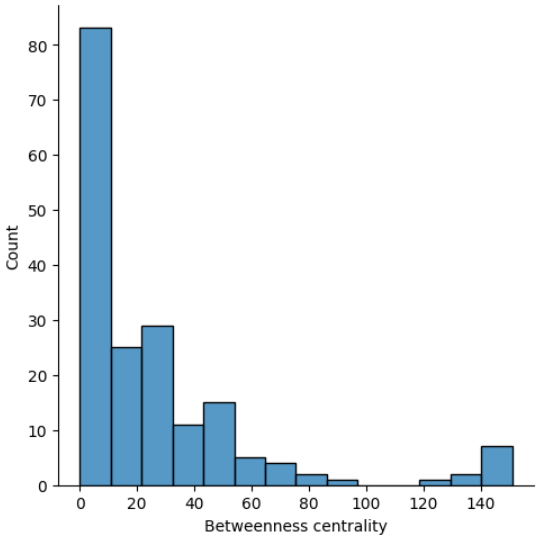


*Hình 2: Biểu đồ thể hiện giá trị bậc của các node*

#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'])



*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'])



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

* 1. Sử dụng Gephi

A graph with red dots

Description automatically generated

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

*A screenshot of a table with numbers

Description automatically generated*

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

*A table with black text

Description automatically generated*

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

1. 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.

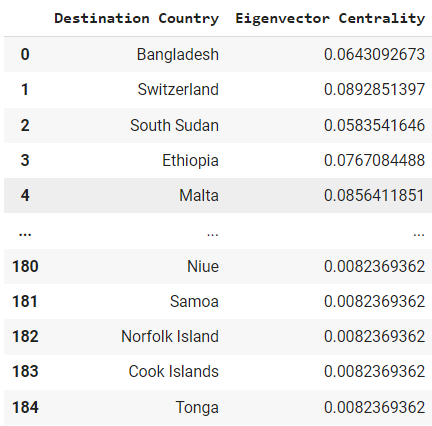
Ý nghĩa: các quốc gia có giá trị eigenvector cao được xem như một quốc gia trọng tâm trong mạng lưới, có mối quan hệ chặt chẽ với các quốc gia khác, là điểm đến của hầu hết các hãng hàng không.

* 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'])

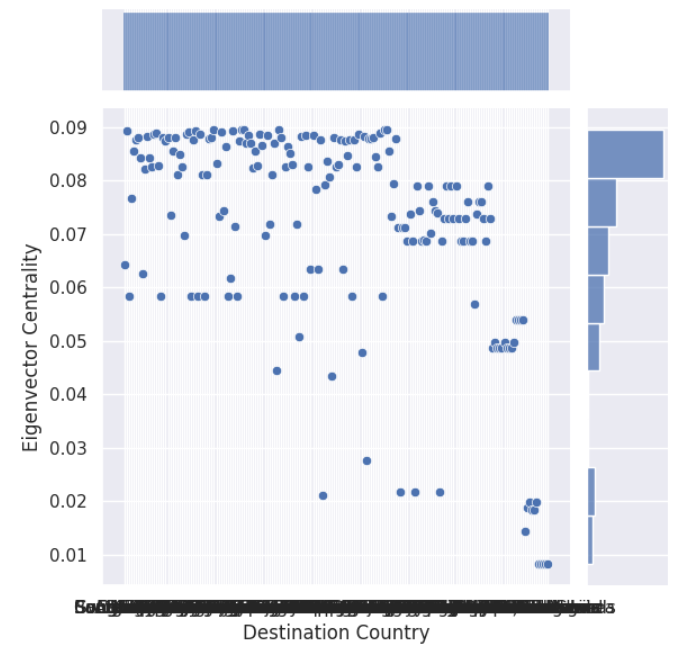
eig\_cen\_df



*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']))

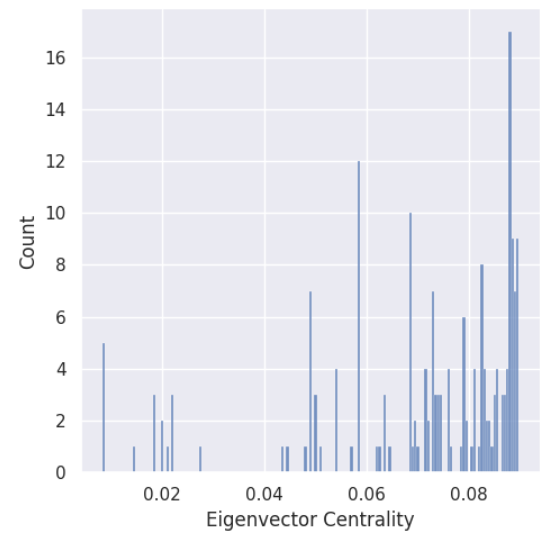
sns.jointplot(x="Destination Country", y="Eigenvector Centrality", data=eig\_cen\_df)



*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)

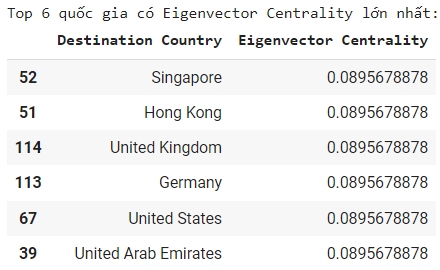
**

*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)

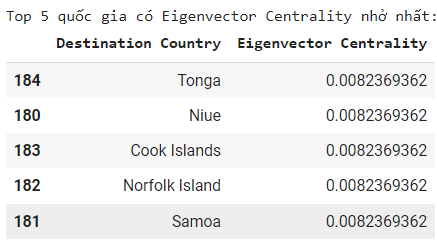
**

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

print('Top 5 quốc gia có Eigenvector Centrality nhỏ nhất: ')

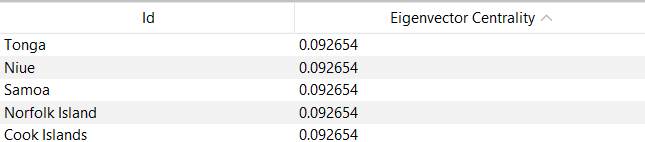
eig\_cen\_df = eig\_cen\_df.sort\_values(by='Eigenvector Centrality', ascending=True)

eig\_cen\_df.head(5)

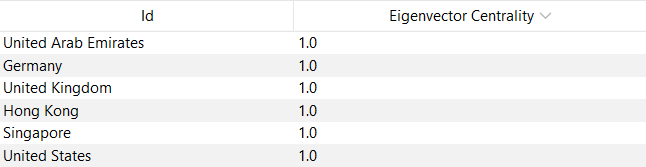
**

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

* 1. Sử dụng Gephi



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

**

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

1. 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.

* 1. Sử dụng Python

pagerank = nx.pagerank(G, tol=1e-6, alpha=0.85)

pagerank\_df = print\_table(pagerank.items(), ['Destination Country', 'Pagerank'])

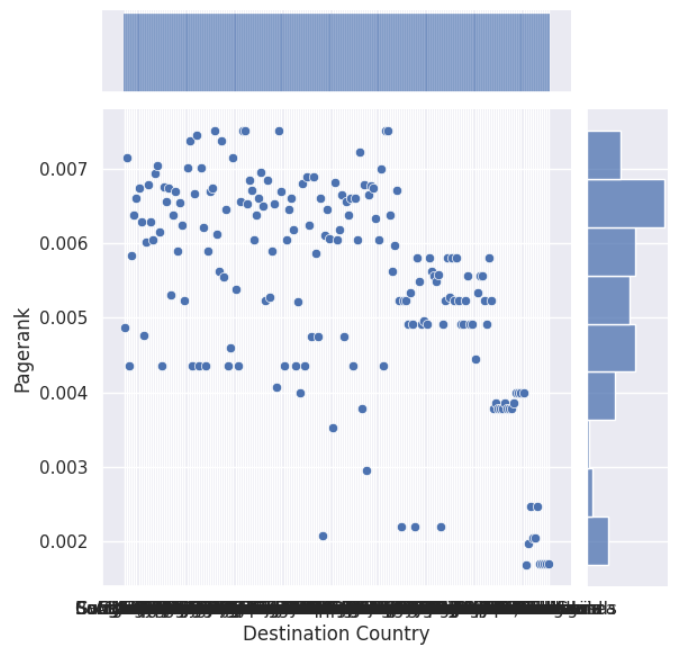
pagerank\_df



*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']))

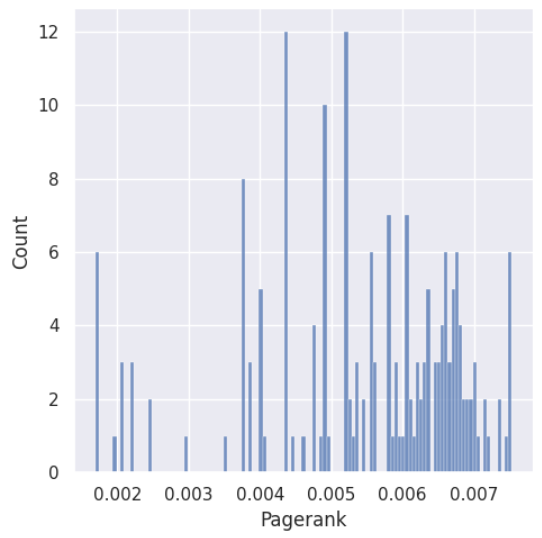
sns.jointplot(x="Destination Country", y="Pagerank", data=pagerank\_df)



*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)

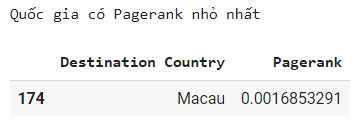


*Hình 4: Liệt kê các quốc gia có giá trị PageRank cao nhất*

print('Quốc gia có Pagerank nhỏ nhất \n')

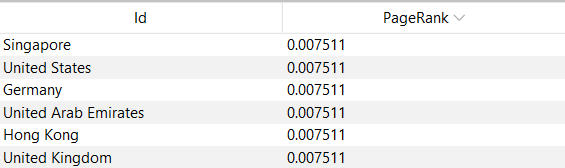
pagerank\_df = pagerank\_df.sort\_values(by='Pagerank', ascending=True)

pagerank\_df.head(1)



*Hình 5: Quốc gia có giá trị PageRank nhỏ nhất*

* 1. Sử dụng Gephi



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

**

*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()

A black and yellow network

Description automatically generated

*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))

A white background with black numbers

Description automatically generated

*Hình 2: Đếm số cụm có trong đồ thị*

cluster\_df = print\_table(partition.items(), ['Node', 'Cluster'])

cluster\_df

A screenshot of a cellphone

Description automatically generated

*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:

1. 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

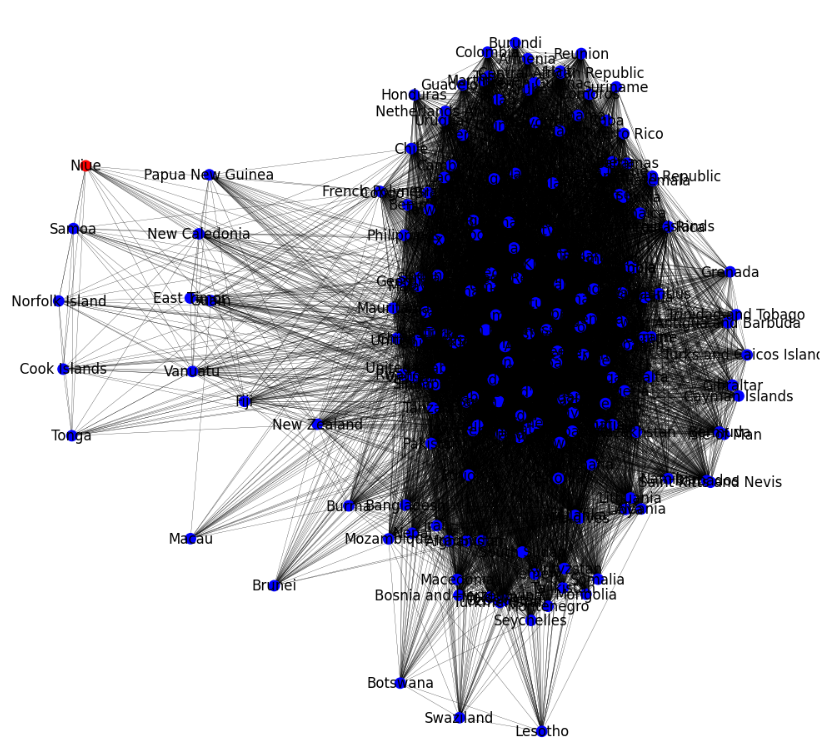
vmin = min(colors)

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:

A network of lines and dots

Description automatically generated

1. Degree Centrality

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

1. Closeness Centrality

A screenshot of a spreadsheet

Description automatically generated

*Kết quả*

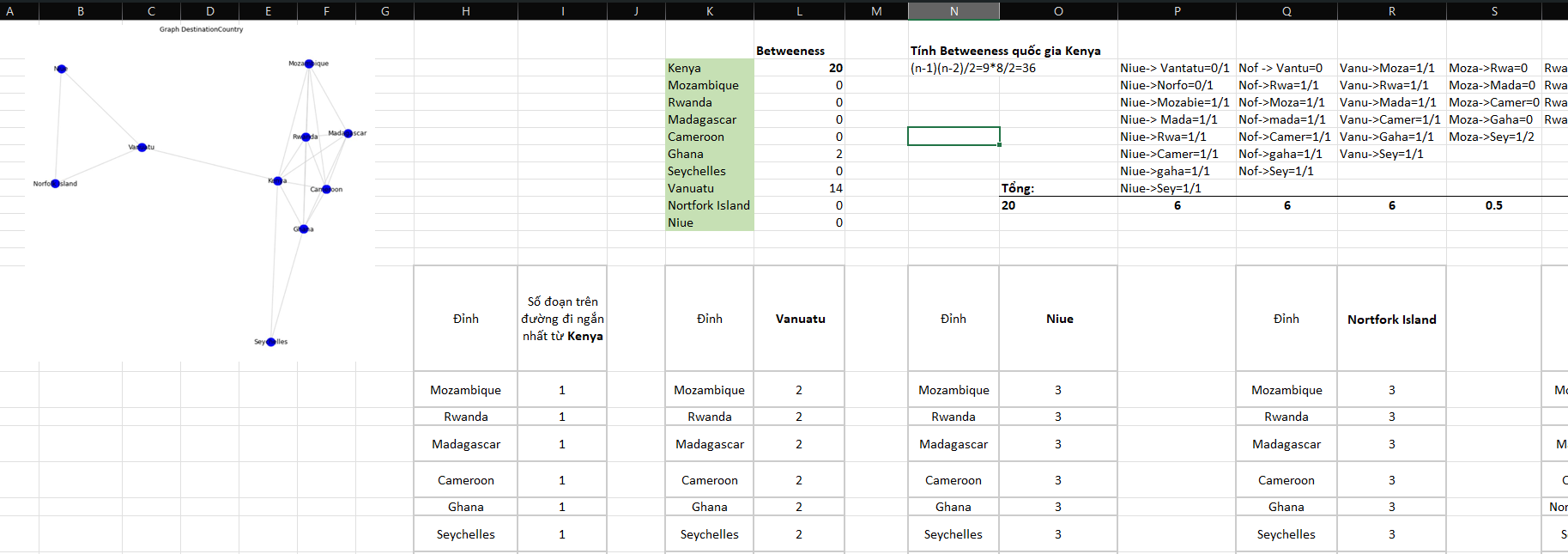
1. Harmonic Centrality

A table with numbers and a number on it

Description automatically generated

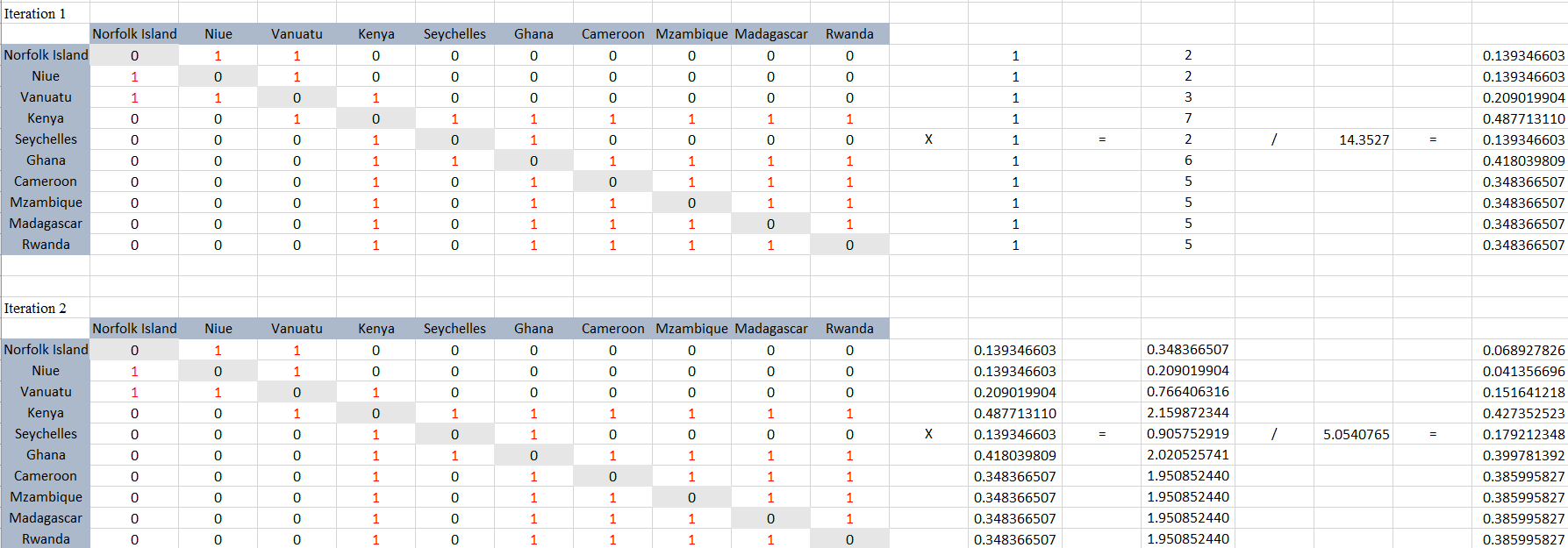
*Kết quả*

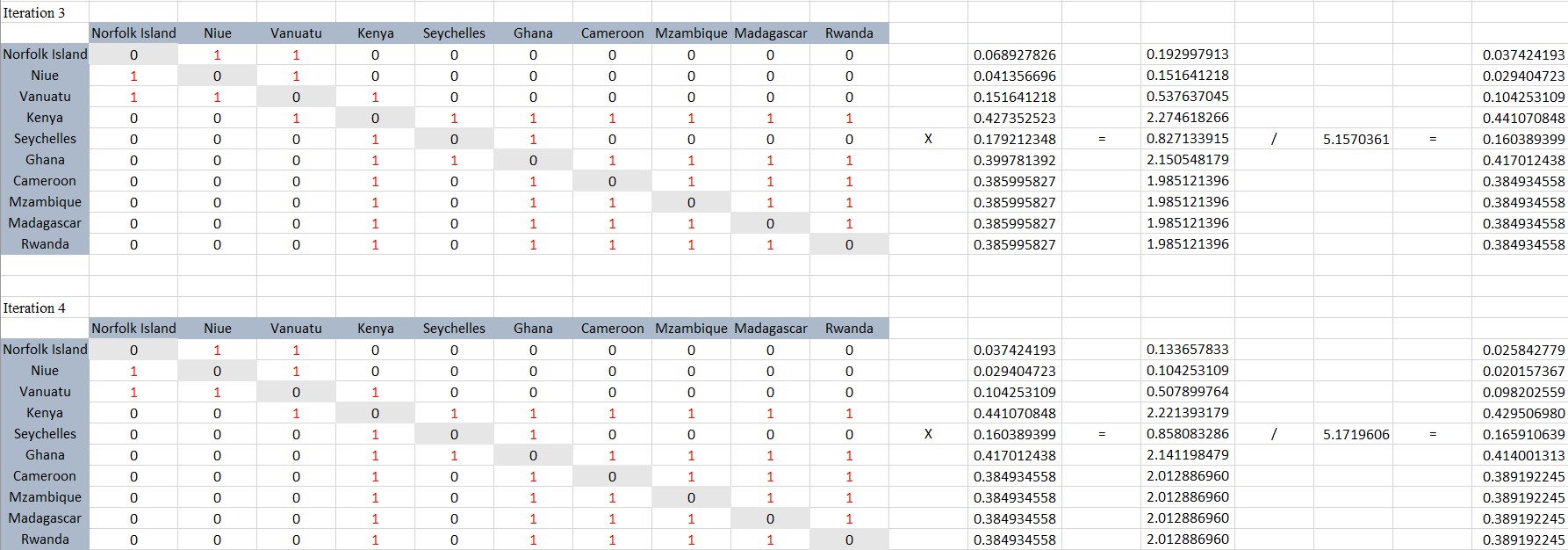
1. Betweenness Centrality

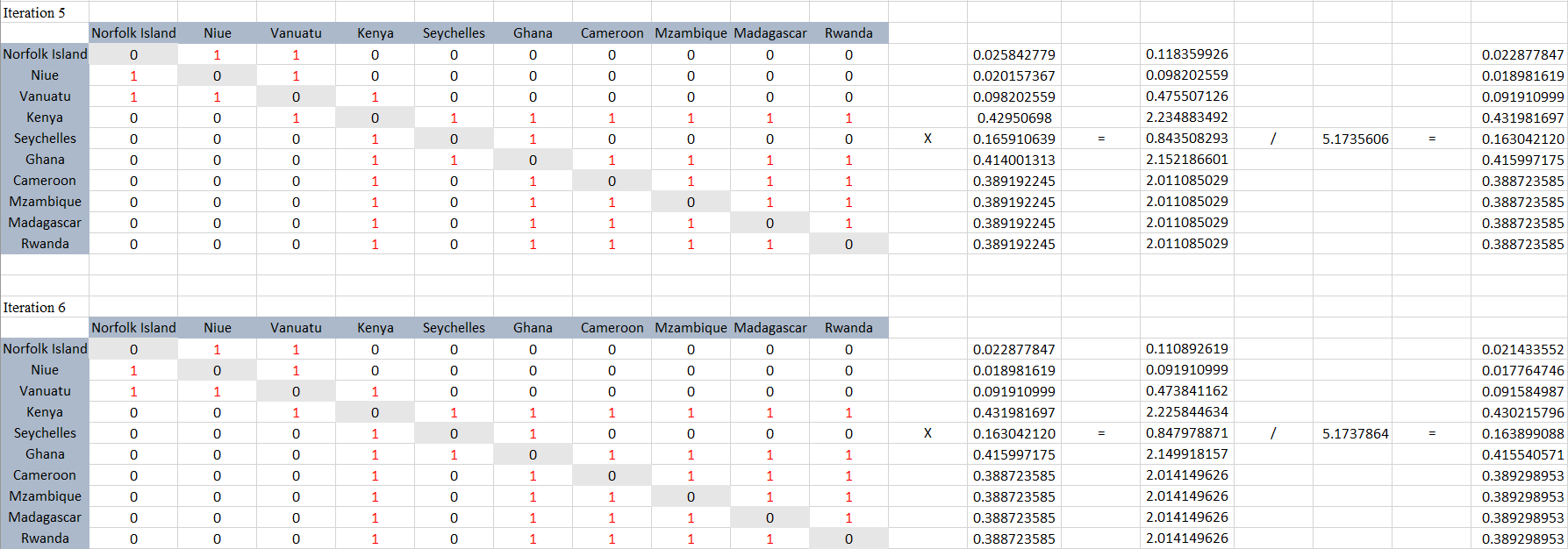
****

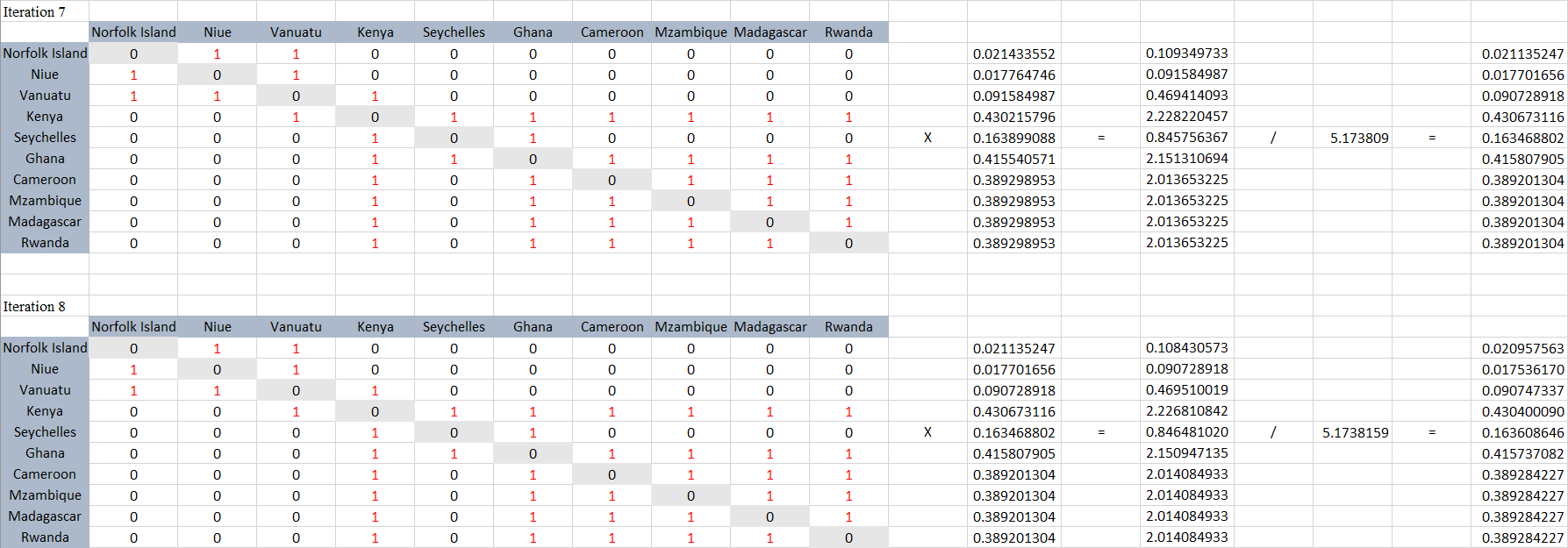
*Kết quả*

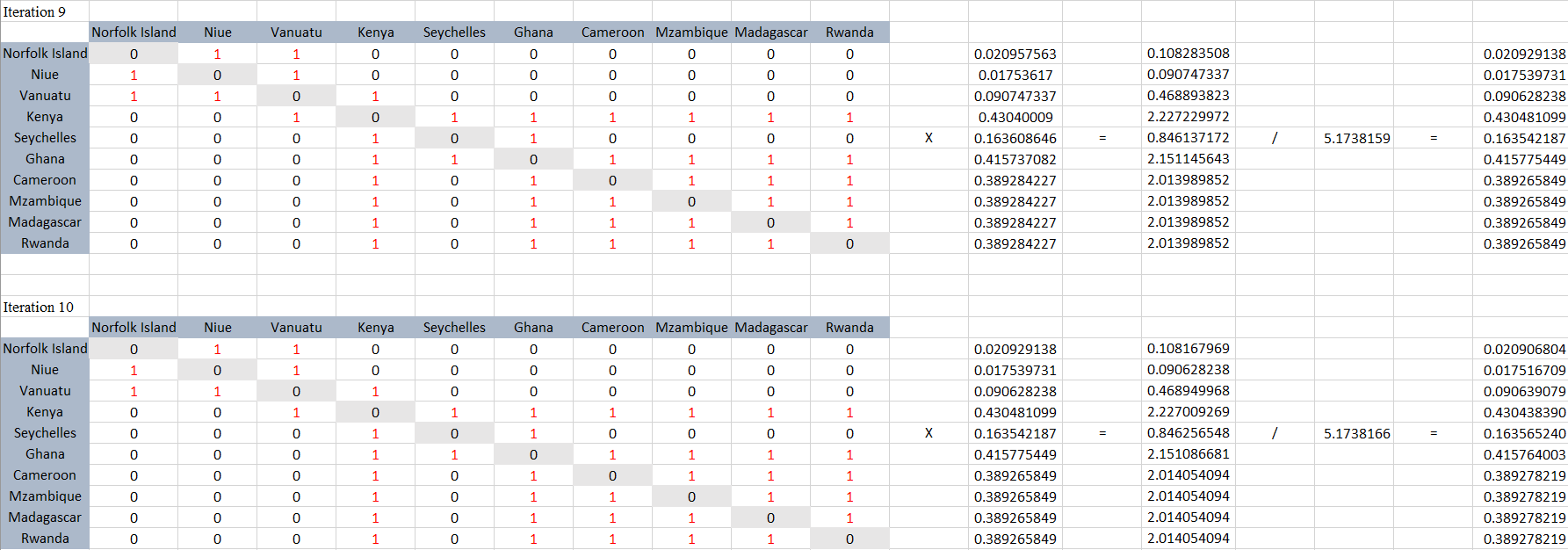
1. Eigenvector Centrality

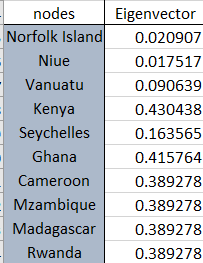
****

****

****

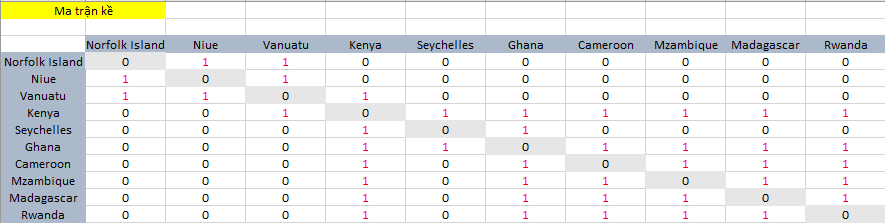
****

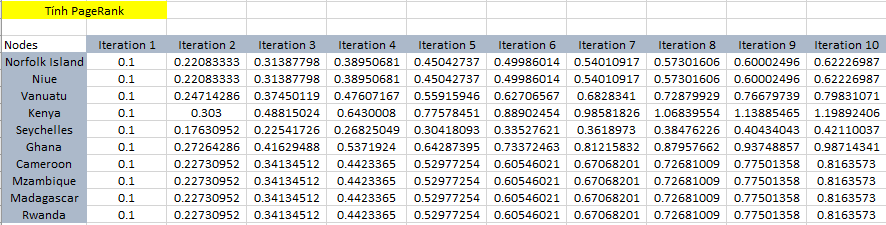
****

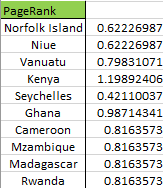
****

*Kết quả*

1. PageRank

****

****

****

*Kết quả*

1. Thuật toán gom cụm Louvain

Ma trận cạnh:

A table with numbers and letters

Description automatically generated

*Hình 1: Ma trận thu được từ đồ thị*

Phân cụm:

A table with numbers and letters

Description automatically generated

*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:

A table with names and numbers

Description automatically generated with medium confidence

*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.

1. 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.

* 1. 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  STT 31 | Nguyễn Anh Nguyên  STT 33 | Lê Thị Đoan Trang  STT 46 | Nguyễn Thị Thảo Trang  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 |