ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

A blue logo with a black background

Description automatically generated with low confidence

ĐỒ ÁN CUỐI KỲ MÔN MẠNG XÃ HỘI

CHỦ ĐỀ DATASET: HỆ THỐNG HÀNG KHÔNG NỘI ĐỊA VÀ THUỘC ĐỊA CỦA MỸ NĂM 2018

**GVHD**: ThS. Nguyễn Thị Kim Phụng

**Sinh viên thực hiện**: Bùi Thị Thanh Ngân - 20521643

TPHCM, tháng 5 năm 2023

# MỤC LỤC HÌNH ẢNH

[Hình 1.1: Danh sách hãng hàng không và mã của các hãng hàng không thuộc các bang/ thuộc địa của Mỹ 7](#_heading=h.tyjcwt)

[Hình 1.2: Danh sách điểm đến là các bang/ thuộc địa của Mỹ và mã các điểm đến 8](#_heading=h.3dy6vkm)

[Hình 1.3: Flights 9](#_heading=h.1t3h5sf)

[Hình 1.4: Airline 10](#_heading=h.4d34og8)

[Hình 1.5: Destination 11](#_heading=h.2s8eyo1)

[Hình 1.6: Dùng SQL Server để kết 3 bảng 12](#_heading=h.17dp8vu)

[Hình 1.7: Data 13](#_heading=h.3rdcrjn)

[Hình 2.1: Code import các thư viện cần thiết 15](#_heading=h.lnxbz9)

[Hình 2.2: Code xóa các dữ liệu trùng lặp 15](#_heading=h.35nkun2)

[Hình 2.3: Code đồ thị hai phía 16](#_heading=h.1ksv4uv)

[Hình 2.4: Code xuất ra đồ thị hai phía 16](#_heading=h.44sinio)

[Hình 2.5: Đồ thị hai phía 17](#_heading=h.2jxsxqh)

[Hình 2.6: Code đồ thị một phía 17](#_heading=h.z337ya)

[Hình 2.7: Đồ thị 1 phía 18](#_heading=h.3j2qqm3)

[Hình 2.8: Code xuất file Gelphi 19](#_heading=h.1y810tw)

[Hình 2.9: Code xuất ra số đỉnh số cạnh và kiểm tra đồ thị liên thông 19](#_heading=h.4i7ojhp)

[Hình 2.10: File Gephi 20](#_heading=h.2xcytpi)

[Hình 2.11: Thực hiện trên Gephi 20](#_heading=h.1ci93xb)

[Hình 3.1: Code Deegree Centrality 21](#_heading=h.3as4poj)

[Hình 3.2: Kết quả Degree Centrality 22](#_heading=h.1pxezwc)

[Hình 3.3: Top 10 node có Degree Centrality cao nhất 23](#_heading=h.49x2ik5)

[Hình 3.4: Top 10 node có Degree Centrality thấp nhất 23](#_heading=h.2p2csry)

[Hình 3.5: Mật độ phân bố Degree Centrality 24](#_heading=h.3o7alnk)

[Hình 3.6: Top 10 node có Degree Centrality cao nhất trên Gephi 24](#_heading=h.23ckvvd)

[Hình 3.7: Top 10 node có Degree Centrality thấp nhất trên Gephi 24](#_heading=h.ihv636)

[Hình 3.8: Code Closeness Centrality 25](#_heading=h.41mghml)

[Hình 3.9: Kết quả Closeness Centrality 25](#_heading=h.2grqrue)

[Hình 3.10: Top 10 node có Closeness Centrality cao nhất 26](#_heading=h.vx1227)

[Hình 3.11: Top 10 node có Closeness Centrality thấp nhất 26](#_heading=h.3fwokq0)

[Hình 3.12: Mật độ phân bố Closeness Centrality 27](#_heading=h.4f1mdlm)

[Hình 3.13: Top 10 node có Closeness Centrality cao nhất trên Gephi 27](#_heading=h.2u6wntf)

[Hình 3.14: Top 10 node có Closeness Centrality thấp nhất trên Gephi 27](#_heading=h.19c6y18)

[Hình 3.15: Code Betweeness Centrality 28](#_heading=h.nmf14n)

[Hình 3.16: Top 10 node có Betweeness Centrality cao nhất 29](#_heading=h.37m2jsg)

[Hình 3.17: Top 10 node có Betweeness Centrality thấp nhất 29](#_heading=h.1mrcu09)

[Hình 3.18: Mật độ phân bố Betweeness Centrality 30](#_heading=h.2lwamvv)

[Hình 3.19: Top 10 node có Betweeness Centrality cao nhất trên Gephi 30](#_heading=h.111kx3o)

[Hình 3.20: Top 10 node có Betweeness Centrality thấp nhất trên Gephi 30](#_heading=h.3l18frh)

[Hình 3.21: Code Eigenvector 31](#_heading=h.2zbgiuw)

[Hình 3.22: Top 10 node có Eigenvector cao nhất 32](#_heading=h.1egqt2p)

[Hình 3.23: Top 10 node có Eigenvector thấp nhất 32](#_heading=h.3ygebqi)

[Hình 3.24: Mật độ phân bố EigenVector Centrality 33](#_heading=h.sqyw64)

[Hình 3.25: Top 10 node có Eigenvector cao nhất trên Gephi 33](#_heading=h.3cqmetx)

[Hình 3.26: Top 10 node có Eigenvector thấp nhất trên Gephi 33](#_heading=h.1rvwp1q)

[Hình 3.27:Code Pagerank 34](#_heading=h.1664s55)

[Hình 3.28: Top 10 node có Pagerank cao nhất 35](#_heading=h.3q5sasy)

[Hình 3.29: Top 10 node có Pagerank thấp nhất 35](#_heading=h.25b2l0r)

[Hình 3.30: Biểu đồ phân bố Pagerank 36](#_heading=h.kgcv8k)

[Hình 3.31: Mật độ phân bố Pagerank 36](#_heading=h.1jlao46)

[Hình 3.32: Top 10 node có Pagerank cao nhất trên Gephi 37](#_heading=h.43ky6rz)

[Hình 3.33: Top 10 node có Pagerank thấp nhất trên Gephi 37](#_heading=h.2iq8gzs)

[Hình 3.34: Code Louvain 37](#_heading=h.1x0gk37)

[Hình 3.35: Kết quả thuật toán Louvain 38](#_heading=h.4h042r0)

[Hình 3.36: Code vẽ biểu đồ phân cụm sau khi dùng Louvain 38](#_heading=h.2w5ecyt)

[Hình 3.37: Biểu đồ phân cụm Louvain 38](#_heading=h.1baon6m)

[Hình 3.38:Cụm 0 (Louvain) 40](#_heading=h.2afmg28)

[Hình 3.39: Cụm 1 (Louvain) 40](#_heading=h.pkwqa1)

[Hình 3.40: Cụm 2 (Louvain) 41](#_heading=h.39kk8xu)

[Hình 3.41: Cụm 3 (Louvain) 41](#_heading=h.1opuj5n)

[Hình 3.42: Copy data 44](#_heading=h.2250f4o)

[Hình 3.43: Encode cột Airline 44](#_heading=h.haapch)

[Hình 3.44: Encode cột Destination 45](#_heading=h.319y80a)

[Hình 3.45: Code dùng Elbow để tìm ra số cụm cho Kmeans 45](#_heading=h.1gf8i83)

[Hình 3.46: Kết quả số cụm của Kmeans khi sử dụng Elbow 45](#_heading=h.40ew0vw)

[Hình 3.47: Dự đoán cụm bằng Kmeans 45](#_heading=h.2fk6b3p)

[Hình 3.48: Kết quả phân cụm Kmeans 46](#_heading=h.upglbi)

[Hình 3.49: Biểu đồ phân cụm Kmeans 46](#_heading=h.3ep43zb)

[Hình 3.50: Cụm 0 (Kmeans) 47](#_heading=h.4du1wux)

[Hình 3.51: Cụm 2 (Kmeans) 47](#_heading=h.2szc72q)

[Hình 3.52: Cụm 2 (Kmeans) 48](#_heading=h.184mhaj)

# MỤC LỤC

[**MỤC LỤC HÌNH ẢNH 2**](#_heading=h.gjdgxs)

[**MỤC LỤC 4**](#_heading=h.30j0zll)

[**CHƯƠNG 1:**](#_heading=h.1fob9te) **GIỚI THIỆU ĐỀ TÀI 6**

[**1.**](#_heading=h.3znysh7) **Giới thiệu 6**

[**2.**](#_heading=h.2et92p0) **Dataset 6**

[**CHƯƠNG 2:**](#_heading=h.26in1rg) **XỬ LÝ DỮ LIỆU 15**

[**CHƯƠNG 3:**](#_heading=h.3whwml4) **CÁC THUẬT TOÁN 21**

[**1.**](#_heading=h.2bn6wsx) **Degree Centrality 21**

[**1.1.**](#_heading=h.qsh70q) **Sử dụng Python 21**

[**1.2.**](#_heading=h.147n2zr) **Sử dụng Gelphi 24**

[**2.**](#_heading=h.32hioqz) **Closeness Centrality 25**

[**2.1.**](#_heading=h.1hmsyys) **Sử dụng Python 25**

[**2.2.**](#_heading=h.1v1yuxt) **Sử dụng Gephi 27**

[**3.**](#_heading=h.3tbugp1) **Betweenness Centrality 28**

[**3.1.**](#_heading=h.28h4qwu) **Sử dụng Python 28**

[**3.2.**](#_heading=h.46r0co2) **Sử dụng Gephi 30**

[**4.**](#_heading=h.206ipza) **EigenVector 31**

[**4.1.**](#_heading=h.4k668n3) **Sử dụng Python 31**

[**4.2.**](#_heading=h.2dlolyb) **Sử dụng Gephi 33**

[**5.**](#_heading=h.4bvk7pj) **Pagerank 34**

[**5.1.**](#_heading=h.2r0uhxc) **Sử dụng Python 34**

[**5.2.**](#_heading=h.34g0dwd) **Sử dụng Gephi 36**

[**6.**](#_heading=h.xvir7l) **Louvain 37**

[**6.1.**](#_heading=h.3hv69ve) **Sử dụng Python 37**

[**6.2.**](#_heading=h.3vac5uf) **Đặc trưng cụm 40**

[**7.**](#_heading=h.48pi1tg) **Girvan Newman 41**

[**8.**](#_heading=h.2nusc19) **Kmeans 43**

[**8.1.**](#_heading=h.3mzq4wv) **Sử dụng Python 44**

[**8.2.**](#_heading=h.1tuee74) **Đặc trưng cụm 47**

[**CHƯƠNG 4:**](#_heading=h.3s49zyc) **TRÍCH XUẤT 10 NODE VÀ THỰC HIỆN TÍNH TAY 48**

# GIỚI THIỆU ĐỀ TÀI

1. **Giới thiệu**

Tầm quan trọng của ngành hàng không hiện nay:

* Kết nối toàn cầu: Ngành hàng không cung cấp phương tiện nhanh chóng và hiệu quả nhất để kết nối các quốc gia và lục địa trên toàn cầu. Điều này giúp thúc đẩy sự phát triển kinh tế, thương mại, du lịch và giao thương quốc tế.
* Vận chuyển hàng hóa quan trọng: Hàng không là phương tiện quan trọng trong vận chuyển hàng hóa như hàng hóa dễ hỏa chất, sản phẩm y tế, thực phẩm tươi sống và hàng hóa giá trị cao. Các chuyến bay hàng hóa giúp đảm bảo chuỗi cung ứng toàn cầu và đáp ứng nhu cầu của thị trường quốc tế.
* Đóng góp vào du lịch: Ngành hàng không đóng góp lớn vào sự phát triển của ngành du lịch. Nó cho phép du khách dễ dàng di chuyển và khám phá các địa điểm trên thế giới, tạo ra thu nhập và việc làm cho nhiều quốc gia.

Và đó cũng là lý do em lựa chọn nghiên cứu đề tài về ngành hàng không để phân tích trong môn mạng xã hội.

Mục tiêu đề tài

Phân tích “mạng lưới phủ sóng” của các hãng hàng không của Hoa Kỳ năm 2018 thông qua các thuật toán xếp hạng, phân cụm như Degree Centrality, Closeness Centrality, Betweeness Centrality, Pagerank,Eigenvector, Louvain, Girvan Newman.

1. **Dataset**

Link dataset: <https://www.kaggle.com/datasets/zernach/2018-airplane-flights>

Dataset bao gồm 14 cột trong đó 2 cột được sử dụng là AirlineCompany và DestWasc.

A picture containing text, screenshot, font

Description automatically generated

*Hình 1.1: Danh sách hãng hàng không và mã của các hãng hàng không thuộc các bang/ thuộc địa của Mỹ*

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 1.2: Danh sách điểm đến là các bang/ thuộc địa của Mỹ và mã các điểm đến*

Data được tổng hợp từ 3 file : Flights, Destination và Airline

A screenshot of a computer

Description automatically generated with low confidence

*Hình 1.3: Flights*

Flight chính là file data gốc được lấy từ link kaggle đã được để ở trên. Flight gồm 14 cột và hơn 1 triệu dòng.

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 1.4: Airline*

Airline là file được tạo từ hình danh sách các hãng hàng không được chú giải trong link dataset. File airline này em tự tạo từ phần chú giải đó và không có trong data gốc. Trong file này gồm 12 hãng hàng không.

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 1.5: Destination*

Destination cũng là file được tạo từ danh sách các điểm đến là các bang/ thuộc địa của nước Mỹ có trong phần giải thích dataset và gồm 54 điểm đến.

Em dùng SQL Server để kết 3 bảng này lại.

A screenshot of a computer

Description automatically generated

*Hình 1.6: Dùng SQL Server để kết 3 bảng*

Do dataset này khá lớn hơn 1 triệu dòng và mỗi một hãng bay gồm khoảng hơn 100 - 200 nghìn dòng và được xếp theo từng hãng do đó để đảm bảo có đầy đủ được tất cả các hãng bay thì với mỗi hãng em sẽ chọn ra 3000 dòng , tổng cộng số dòng mà em có được là 21000 dòng với 2 cột là tên hãng hàng không và điểm đến.

A screenshot of a computer

Description automatically generated with low confidence

*Hình 1.7: Data*

Đây là file data em có được sau khi kết 3 bảng. Bao gồm 2 cột và 21000 dòng với 7 hãng hàng không và 40 điểm đến.

Bài toán cần giải quyết: Dựa trên dataset trên, chọn ra các đỉnh, cạnh và áp dụng các thuật toán mạng xã hội để tìm ra bang/ thuộc địa dễ đi đến bằng đường hàng không nhất.

Đề xuất mô hình xử lý:

Lấy các bang/ thuộc địa làm **ĐỈNH**

**CẠNH** là đường nối giữa hai bang/thuộc địa mà chúng cùng có một hãng hàng không ( cùng là điểm đến của một hãng hàng không).

# XỬ LÝ DỮ LIỆU

Import các thư viện cần thiết

A picture containing text, screenshot, font

Description automatically generated

*Hình 2.1: Code import các thư viện cần thiết*

Đọc dữ liệu từ file csv

Xóa các dữ liệu trống và dữ liệu trùng lặp

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 2.2: Code xóa các dữ liệu trùng lặp*

Xuất ra số hãng hàng không và số điểm đến, số cạnh

A computer code on a white background

Description automatically generated with low confidence

Vẽ đồ thị hai phía

Node: là tên của hãng hàng không, các điểm đến là các bang/ thuộc địa của Mỹ

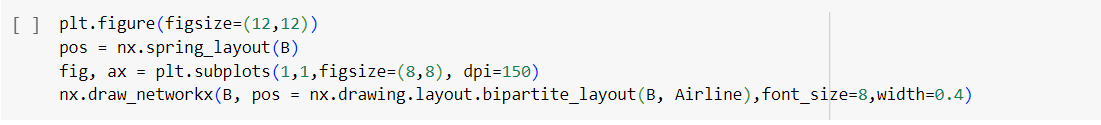
Edge: mối quan hệ giữa các điểm đến của các hãng hàng không

A picture containing text, font, line, screenshot

Description automatically generated

*Hình 2.3: Code đồ thị hai phía*

Xuất ra đồ thị hai phía



*Hình 2.4: Code xuất ra đồ thị hai phía*

A picture containing line, diagram, screenshot

Description automatically generated

*Hình 2.5: Đồ thị hai phía*

Quan sát đồ thị có thể thấy được một hãng hàng không có rất nhiều điểm đến là các bang/thuộc địa và một điểm đến thì được phủ sóng bởi rất nhiều hãng hàng không.

A computer code on a white background

Description automatically generated with low confidence

*Hình 2.6: Code đồ thị một phía*

Đồ thị một phía

Node: là các điểm đến là các bang/ thuộc địa thuộc nước Mỹ

Edge: Hai bang/thuộc địa cùng là điểm đến của 1 hãng hàng không sẽ hợp thành 1 cạnh

Điều này nói lên rằng mỗi hàng hàng không vận chuyển khách tới nhiều bang/thuộc địa khác nhau, tại mỗi bang/thuộc địa có sự cạnh tranh giữa các hãng hàng không.

Weight: Trọng số là số bang/thuộc địa cùng là điểm đến của các hãng hàng không

A picture containing diagram, line, origami

Description automatically generated

*Hình 2.7: Đồ thị 1 phía*

Lưu các cạnh của đồ thị vào edges

Tạo dataframe là các cạnh gồm : u,v,weight

Xuất ra file Gelphi để đưa vào phần mềm Gelphi

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 2.8: Code xuất file Gelphi*

Xuất ra số đỉnh và số cạnh của đồ thị và kiểm tra đồ thị liên thông.

A close-up of a computer screen

Description automatically generated with low confidence

*Hình 2.9: Code xuất ra số đỉnh số cạnh và kiểm tra đồ thị liên thông*

u và v là 2 node của cạnh, là 2 địa điểm cùng là điểm đến của ít nhất 1 hãng hàng không.

Weight: Trọng số, số hãng hàng không chọn cả 2 địa điểm u v làm điểm đến.

Sau khi xuất được file Gephi em chỉ sử dụng hai cột là u và v và đổi tên lại thành source target để import vào Gephi.

A screenshot of a spreadsheet

Description automatically generated

*Hình 2.10: File Gephi*

A picture containing screenshot

Description automatically generated

*Hình 2.11: Thực hiện trên Gephi*

# CÁC THUẬT TOÁN

1. **Degree Centrality**

Độ trung tâm cấp bậc (Degree centrality) của môṭ nút là tổng số lươṇ g các liên kết thưc̣ tế của nút đó với các nút khác trong mạng lưới . Người ta cũng có thể phân loaji số lượng liên kết của môṭ nút về mức độ trung tâm với hai loại là mức độ đi vào (in-degree) và mức độ đi ra (outdegree). Theo đó, mức độ đi vào là tổng số lượng các liên kết xuất phát từ các nút khác tới nút đang xem xét và mức độ đi ra là số lượng các liên kết trực tiếp từ nút đó đến những nút khác trong mạng lưới.

* 1. **Sử dụng Python**

Code Degree Centrality

A picture containing text, font, screenshot

Description automatically generated

*Hình 3.1: Code Deegree Centrality*

Kết quả Degree Centrality

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.2: Kết quả Degree Centrality*

Top 10 node có Degree Centrality cao nhất

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.3: Top 10 node có Degree Centrality cao nhất*

Top 10 node có Degree Centrality thấp nhất

A screenshot of a computer

Description automatically generated

*Hình 3.4: Top 10 node có Degree Centrality thấp nhất*

* 1. **Sử dụng Gelphi**

A graph with red dots

Description automatically generated with low confidence

*Hình 3.5: Mật độ phân bố Degree Centrality*

Top 10 node có Degree cao nhất

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.6: Top 10 node có Degree Centrality cao nhất trên Gephi*

Top 10 node có Degree thấp nhất

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.7: Top 10 node có Degree Centrality thấp nhất trên Gephi*

1. **Closeness Centrality**
   1. **Sử dụng Python**

Closeness centrality là một khái niệm trong mạng lưới (network) và phân tích đồ thị (graph analysis) để đo đạc tầm quan trọng của một đỉnh (node) trong mạng dựa trên tầm xa trung bình giữa đỉnh đó và tất cả các đỉnh khác trong mạng. Nó đo lường khả năng tiếp cận của một đỉnh trong mạng, tức là thời gian hoặc số lượng bước cần thiết để đi từ đỉnh đó đến tất cả các đỉnh khác trong mạng.

Closeness centrality có thể được sử dụng để xác định các đỉnh quan trọng trong mạng, như những đỉnh có thể truyền thông nhanh chóng thông tin đến các đỉnh khác hoặc có thể tương tác với một số lượng lớn đỉnh trong mạng.

Code Closeness Centrality

A picture containing text, font, line, screenshot

Description automatically generated

*Hình 3.8: Code Closeness Centrality*

Kết quả Closeness Centrality

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.9: Kết quả Closeness Centrality*

Top 10 node có Closeness Centrality cao nhất

A screenshot of a computer

Description automatically generated

*Hình 3.10: Top 10 node có Closeness Centrality cao nhất*

Top 10 nodes có Closeness Centrality thấp nhất

A screenshot of a computer

Description automatically generated

*Hình 3.11: Top 10 node có Closeness Centrality thấp nhất*

* 1. **Sử dụng Gephi**

A picture containing text, number, font, screenshot

Description automatically generated

*Hình 3.12: Mật độ phân bố Closeness Centrality*

Top 10 node có Closeness Centrality cao nhất

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.13: Top 10 node có Closeness Centrality cao nhất trên Gephi*

Top 10 node có Closeness Centrality thấp nhất

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.14: Top 10 node có Closeness Centrality thấp nhất trên Gephi*

Kết quả Closeness Centrality khi sử dụng Python và Gephi là như nhau.

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 dung 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.

* 1. **Sử dụng Python**

Code và kết quả Betweenness Centrality

A screenshot of a computer

Description automatically generated

*Hình 3.15: Code Betweeness Centrality*

Top 10 node có Betweenness Centrality cao nhất

A screenshot of a computer

Description automatically generated

*Hình 3.16: Top 10 node có Betweeness Centrality cao nhất*

Top 10 node có Betweenness Centrality thấp nhất

A screenshot of a computer

Description automatically generated

*Hình 3.17: Top 10 node có Betweeness Centrality thấp nhất*

* 1. **Sử dụng Gephi**

A graph with red dots

Description automatically generated with low confidence

*Hình 3.18: Mật độ phân bố Betweeness Centrality*

Top 10 node có Betweenness Centrality cao nhất

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.19: Top 10 node có Betweeness Centrality cao nhất trên Gephi*

Top 10 Node có Betweenness Centrality thấp nhất

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.20: Top 10 node có Betweeness Centrality thấp nhất trên Gephi*

1. **EigenVector**
   1. **Sử dụng Python**

Trong mạng xã hội, thuật toán eigenvector có thể được sử dụng để tính toán sự quan trọng của các thành viên trong mạng dựa trên cấu trúc liên kết giữa họ. Thuật toán eigenvector có thể áp dụng để xác định các thành viên quan trọng nhất trong mạng, cũng như tìm ra các nhóm (community) có sự tương tác mạnh với nhau.

Code EigenVector

A screenshot of a computer

Description automatically generated

*Hình 3.21: Code Eigenvector*

Top 10 node có Eigenvector cao nhất

A screenshot of a computer

Description automatically generated

*Hình 3.22: Top 10 node có Eigenvector cao nhất*

Top 10 node có Eigenvector thấp nhất

A screenshot of a computer

Description automatically generated

*Hình 3.23: Top 10 node có Eigenvector thấp nhất*

* 1. **Sử dụng Gephi**

A picture containing text, screenshot, number, font

Description automatically generated

*Hình 3.24: Mật độ phân bố EigenVector Centrality*

Top 10 node có Eigenvector cao nhất

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.25: Top 10 node có Eigenvector cao nhất trên Gephi*

Top 10 node có Eigenvector thấp nhất

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.26: Top 10 node có Eigenvector thấp nhất trên Gephi*

Về giá trị Eigenvector có khác nhau giữa python code và Gephi tuy nhiên thì top 10 node có giá trị Eigenvector cao và thấp nhất không có sự thay đổi.

1. **Pagerank**

PageRank là một thuật toán đo lường ảnh hưởng bắc cầu hoặc kết nối của các nút.

Nó có thể được tính toán bằng cách phân phối lặp đi lặp lại thứ hạng của một nút (ban đầu dựa trên mức độ) cho các nút lân cận hoặc bằng cách duyệt ngẫu nhiên biểu đồ và đếm tần suất chạm vào mỗi nút trong những bước này.

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

* 1. **Sử dụng Python**

Code Pagerank

A screenshot of a computer

Description automatically generated

*Hình 3.27:Code Pagerank*

Top 10 node có Pagerank cao nhất

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.28: Top 10 node có Pagerank cao nhất*

Top 10 node có Pagerank thấp nhất

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.29: Top 10 node có Pagerank thấp nhất*

Biểu đồ phân bố Pagerank

A picture containing text, screenshot, diagram, plot

Description automatically generated

*Hình 3.30: Biểu đồ phân bố Pagerank*

* 1. **Sử dụng Gephi**

A picture containing text, screenshot, number, font

Description automatically generated

*Hình 3.31: Mật độ phân bố Pagerank*

Top 10 node có Pagerank cao nhất

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.32: Top 10 node có Pagerank cao nhất trên Gephi*

Top 10 node có Pagerank thấp nhất

A screenshot of a computer

Description automatically generated with low confidence

*Hình 3.33: Top 10 node có Pagerank thấp nhất trên Gephi*

1. **Louvain**

Thuật toán Louvain là một 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.

* 1. **Sử dụng Python**

A screenshot of a computer program

Description automatically generated with low confidence

*Hình 3.34: Code Louvain*

Kết quả thuật toán Louvain

A screenshot of a computer

Description automatically generated

*Hình 3.35: Kết quả thuật toán Louvain*

A computer code on a white background

Description automatically generated with low confidence

*Hình 3.36: Code vẽ biểu đồ phân cụm sau khi dùng Louvain*

A picture containing map, drawing, screenshot, diagram

Description automatically generated

*Hình 3.37: Biểu đồ phân cụm Louvain*

Cụm 0 #8000ff

A picture containing screenshot, text, purple, lilac

Description automatically generated

--------------------------------------

Cụm 1 #2adddd

A picture containing text, screenshot, rectangle

Description automatically generated

--------------------------------------

Cụm 2 #d4dd80

A green square with black lines

Description automatically generated with low confidence

--------------------------------------

Cụm 3 #ff0000

A red square with black lines

Description automatically generated with low confidence

--------------------------------------

* 1. **Đặc trưng cụm**

*Hình 3.38:Cụm 0 (Louvain)*

*Hình 3.39: Cụm 1 (Louvain)*

*Hình 3.40: Cụm 2 (Louvain)*

*Hình 3.41: Cụm 3 (Louvain)*

Kết quả phân cụm Louvain chia thành 4 cụm.

* Đối với cụm 0: hãng Delta Air Lines Inc có số lượng nhiều nhất 8.
* Đối với cụm 1: hãng SouthWest Ailines Co là nhiều nhất 8.
* Đối với cụm 2: hãng Alaska Airline Inc là nhiều nhất 4.
* Đối với cụm 3: hãng United Air Lines Inc là nhiều nhất 12.

1. **Girvan Newman**

Girvan Newman là thuật toán tìm và loại bỏ các cạnh nối các cộng đồng

trong mạng, các cạnh đóng vai trị ‘trung gian’ trong mạng lưới. 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.

Code remove edge

A screen shot of a computer code

Description automatically generated with low confidence

Code define hàm Girvan Newman

A screenshot of a computer program

Description automatically generated with low confidence

Code find Communities

A picture containing text, screenshot, font

Description automatically generated

Code Girvan Newman

A screenshot of a computer

Description automatically generated with low confidence

Kết quả thuật toán Girvan Newman:

A picture containing text, font, screenshot, line

Description automatically generated

Dựa vào kết quả phân cụm có thể thấy được Cụm 2 chỉ gồm Hawaii và Alaska trong khi cụm 1 bao gồm tất cả các điểm đến khác.

1. **Kmeans**

K-means là một thuật toán phân cụm đơn giản thuộc loại học không giám sát(tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm. Ý tưởng của thuật toán phân cụm k-means là phân chia 1 bộ dữ liệu thành các cụm khác nhau. Trong đó số lượng cụm được cho trước là k. Công việc phân cụm được xác lập dựa trên nguyên lý: Các điểm dữ liệu trong cùng 1 cụm thì phải có cùng 1 số tính chất nhất định. Tức là giữa các điểm trong cùng 1 cụm phải có sự liên quan lẫn nhau. Đối với máy tính thì các điểm trong 1 cụm đó sẽ là các điểm dữ liệu gần nhau.

Thuật toán phân cụm k-means thường được sử dụng trong các ứng dụng cỗ máy tìm kiếm, phân đoạn khách hàng, thống kê dữ liệu,…

* 1. **Sử dụng Python**

Copy data từ df sang dt.

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.42: Copy data*

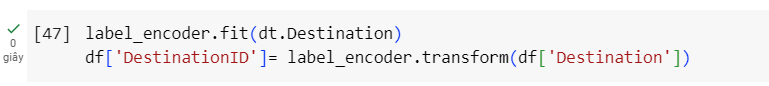
import preprocessing và chuyển đổi cột Airline thành encode

A picture containing text, font, line, screenshot

Description automatically generated

*Hình 3.43: Encode cột Airline*

Chuyển đổi cột Destination thành encode



*Hình 3.44: Encode cột Destination*

Dùng phương pháp elbow để tìm ra số cụm cho thuật toán Kmeans. Số cụm là 3.

A screenshot of a computer program

Description automatically generated with medium confidence

*Hình 3.45: Code dùng Elbow để tìm ra số cụm cho Kmeans*

A graph with a line

Description automatically generated with low confidence

*Hình 3.46: Kết quả số cụm của Kmeans khi sử dụng Elbow*

Sử dụng thuật toán Kmeans với số cụm là 3 vừa tìm được ở trên để dự đoán các cụm

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.47: Dự đoán cụm bằng Kmeans*

Xuất ra các cụm

A screenshot of a computer

Description automatically generated with medium confidence

*Hình 3.48: Kết quả phân cụm Kmeans*

Biểu đồ phân cụm của Kmeans

A screen shot of a computer screen

Description automatically generated with low confidence

*Hình 3.49: Biểu đồ phân cụm Kmeans*

* 1. **Đặc trưng cụm**

*Hình 3.50: Cụm 0 (Kmeans)*

*Hình 3.51: Cụm 2 (Kmeans)*

*Hình 3.52: Cụm 2 (Kmeans)*

Ở thuật toán phân cụm Kmeans ta có thể thấy kết quả phân thành 3 cụm tuy nhiên đặc trưng của 3 cụm không quá nổi bật và giống nhau bởi ở cả ba cụm hãng Delta Air Lines Inc đều chiếm số lượng nhiều nhất.

# TRÍCH XUẤT 10 NODE VÀ THỰC HIỆN TÍNH TAY

Thực hiện trong file Manual Calculation đã nộp trên gg drive