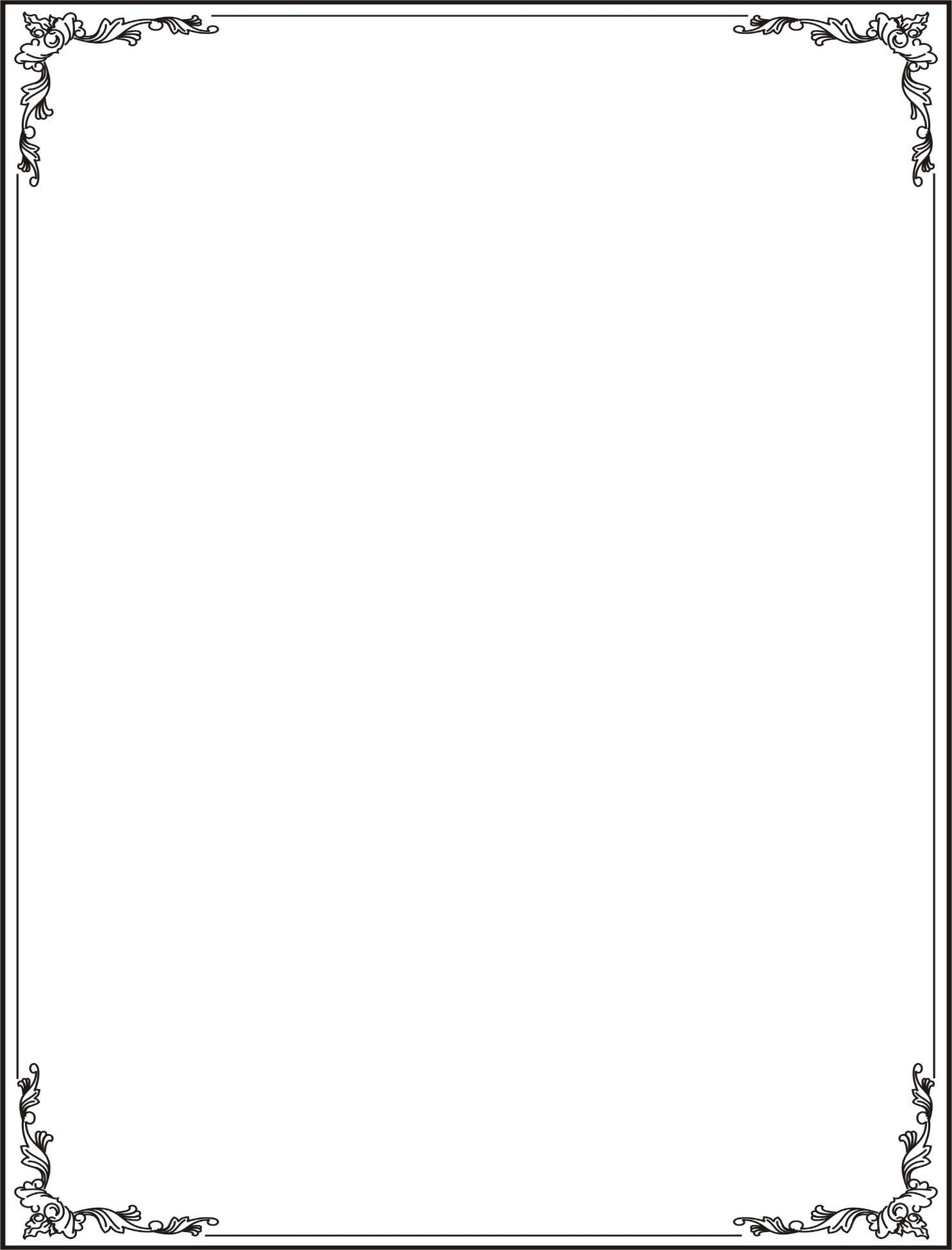
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----



**ĐỒ ÁN CUỐI KÌ**

**Đề tài: Giai thưởng Oscar năm 1927**

**Môn học: Mạng xã hội**

**Giao viên: ThS. *Nguyễn Thị Kim Phụng***

**Lớp: *IS353.N11.HTCL***

**Sinh viên trình bày:**

***Nguyễn Thanh Sang – 19522123***

GIÁO VIÊN NHẬN XÉT

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

**……………………………………………………………………………**

MỤC LỤC

[I. Gioi thiệu 1](#_Toc106174876)

[II. Xác định bài toán 2](#_Toc106174879)

[III. Dữ liệu 2](#_Toc106174885)

[1. Gioi thiệu nguồn dữ liệu 2](#_Toc106174880)

[2. Xử lý và phân tích dữ liệu 3](#_Toc106174880)

[a. Làm sạch dữ liệu 3](#_Toc106174880)

[b. Chuyển đổi dataframe thành đồ thị 4](#_Toc106174881)

[IV. Thuật toán 10](#_Toc106174891)

[1. Thuật toán phát hiện cộng đồng 10](#_Toc106174881)

[a. Thực thi thuật toán Louvain bằng Gephi 10](#_Toc106174880)

[b. Thực thi bằng Python 10](#_Toc106174881)

[2. Centralities 15](#_Toc106174880)

[a. Thực thi thuật toán Betweenness Centrality 15](#_Toc106174880)

[b. Thực thi thuật toán Closeness Centrality 17](#_Toc106174881)

[c. Thực thi thuật toán Eigenvector Centrality 18](#_Toc106174881)

[d. Thực thi thuật toán Page Rank 19](#_Toc106174881)

[3. Graph Mining 21](#_Toc106174881)

[b. Thực thi thuật toán Girvan Newman 21](#_Toc106174880)

[V. Tính tay thuật toán 23](#_Toc106174891)

[1. Centralities 23](#_Toc106174880)

[a. Thực thi thuật toán Betweenness Centrality 23](#_Toc106174880)

[b. Thực thi thuật toán Closeness Centrality 24](#_Toc106174881)

[c. Thực thi thuật toán Page Rank 24](#_Toc106174881)

[2. Clustering 24](#_Toc106174881)

[a. Thực thi thuật toán Louvain 24](#_Toc106174881)

[VI. Tài liệu tham khảo 25](#_Toc106174902)

# **I. GIOI THIỆU**

Ngày nay, nhu cầu giải trí là vô cùng thiết yếu trong cuộc sống đầy áp lực và bộn bề này. Đặc biệt là nhu cầu xem phim ảnh. Nhằm đánh giá chất lượng của bộ phim, giải thưởng Oscar đã ra đời nhằm vinh danh những bộ phim, những diễn viên, đạo diễn,… đã cống hiến những giá trị hữu ích trong giải trí thế giới nói chung và nền điện ảnh thế giới nói riêng. Không những vậy, Oscar còn là một trong những giải thưởng danh giá mà bất kì ekip làm phim nào cũng đều muốn đoạt được trong chặng đường làm phim của mình. Thấy được tầm quan trọng cũng như độ phủ sóng của giải thưởng này, em đã áp dụng kiến thức môn Mạng xã hội để triển khai các thuật toán nhằm phân tích, tìm ra đáp án cho bài toán: “Bộ phim nào được nhiều để cử trong năm 1927”. Sở dĩ em chọn chỉ có năm 1927 vì dữ liệu quá lớn, không thể phân tích được hết. Thứ hai là chỉ có năm 1927 mới có thể thực hiện được đồ thị một phía (vì số lượng đỉnh của dữ liệu sau khi lọc ra phù hợp với số lượng đỉnh của đồ thị).

# **II. XÁC ĐỊNH BÀI TOÁN**

# - Input: Tập dữ liệu The Oscar Award, 1927 - 2020 trên nguồn dữ liệu Kaggle đã qua tiền xử lý dữ liệu

# - Ouput: Đưa ra độ đo, đưa ra cộng đồng phục vụ cho việc phân tích mạng xã hội “The Oscar Award in 1927”

- Bài toán cần giải quyết: Bộ phim nào được nhiều đề cử trong năm 1927?

- Đề xuất mô hình xử lý:

+Clustering: sử dụng thuật toán Girvan Newman, Louvain, K-mean để phân cụm từ đó rút ra ý nghĩa cụm

+Centralities: sử dụng thuật toán Betweenness Centrality, Closeness Centrality, Eigenvector Centrality, Pagerank để giải quyết bài toán.

- Đỉnh là tên phim.

- Cạnh là đường nối giữa tên phim, cạnh thể hiện 2 phim đều nhận được cùng 1 đề cử.

# **III. DỮ LIỆU**

## **1. Gioi thiệu nguồn dữ liệu**

- Link dataset: <https://www.kaggle.com/datasets/unanimad/the-oscar-award>

- Dữ liệu The Oscar Award, 1927 - 2020 được cung cấp trên nền tảng Kaggle chứa dữ liệu các bộ phim đoạt giải Oscar từ năm 1927 đến 2020. Dữ liệu gồm 10396 dòng với 7 thuộc tính bao gồm:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Mô tả | Các giá trị |
| 1 | Year\_film | int | Năm mà phim được chiếu |  |
| 2 | Year\_ceremony | int | Năm mà phim được nhận giải Oscar |  |
| 3 | Ceremony | int | Số buổi lễ |  |
| 4 | Category | Char | Loại đề cử |  |
| 5 | Name | Char | Tên của người được nhận đề cử |  |
| 6 | Film | Char | Tên phim |  |
| 7 | Winner | Char | Chiến thắng đề cử | TRUE: có  FALSE: không |

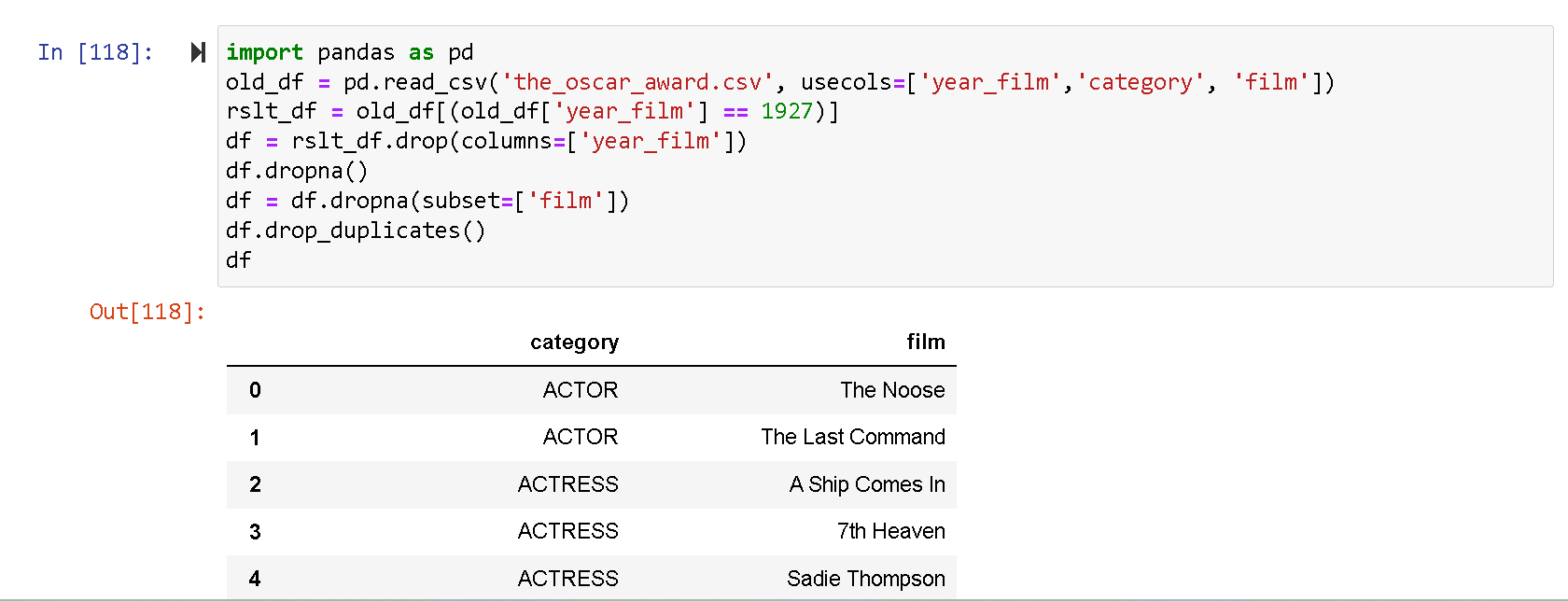
## **2. Xử lý và phân tích dữ liệu**

## **a. Làm sạch dữ liệu**

- Dữ liệu sẽ được đọc từ file csv đưa vào dataframe.

- Với bài toán được đặt ra ở trên thì em chỉ sử dụng cột year\_film, category và film, còn những cột còn lại tiến hành loại bỏ khỏi dataset.

- Vì bài toán em chỉ phân tích những bộ phim đoạt giải Oscar năm 1927, nên em chọn ra những bộ phim được chiếu vào năm 1927.



*Hình 1. Đọc dữ liệu từ file csv và loại bỏ các cột không dùng, dữ liệu trùng và rỗng*

- Sau khi đã lọc xong, em loại bỏ cột year\_film và những dòng có dữ liệu trống hoặc trùng lặp. Kết quả nhận được là một bộ dữ liệu 31 dòng và 2 cột.

## **Chuyển đổi dataframe thành đồ thị**

Đồ thị 2 phía

-Node: là tên phim

-Edge: mối quan hệ giữa đề cử và phim.

**

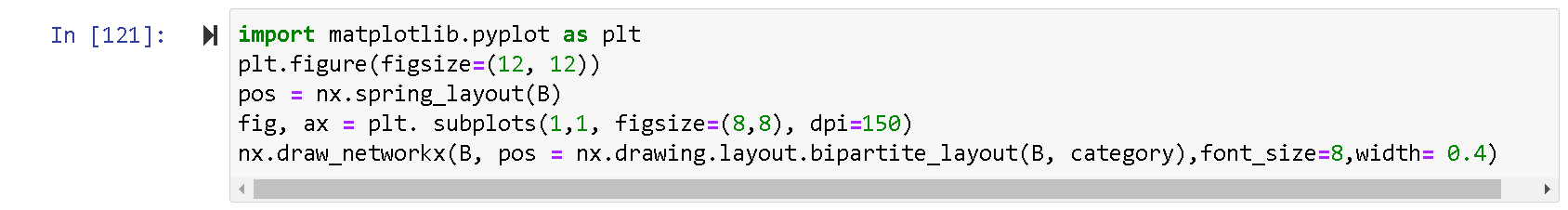
*Hình 2. Đưa dữ liệu từ dataframe vào đồ thị vô hướng*

- Có thể thấy có 12 đề cử

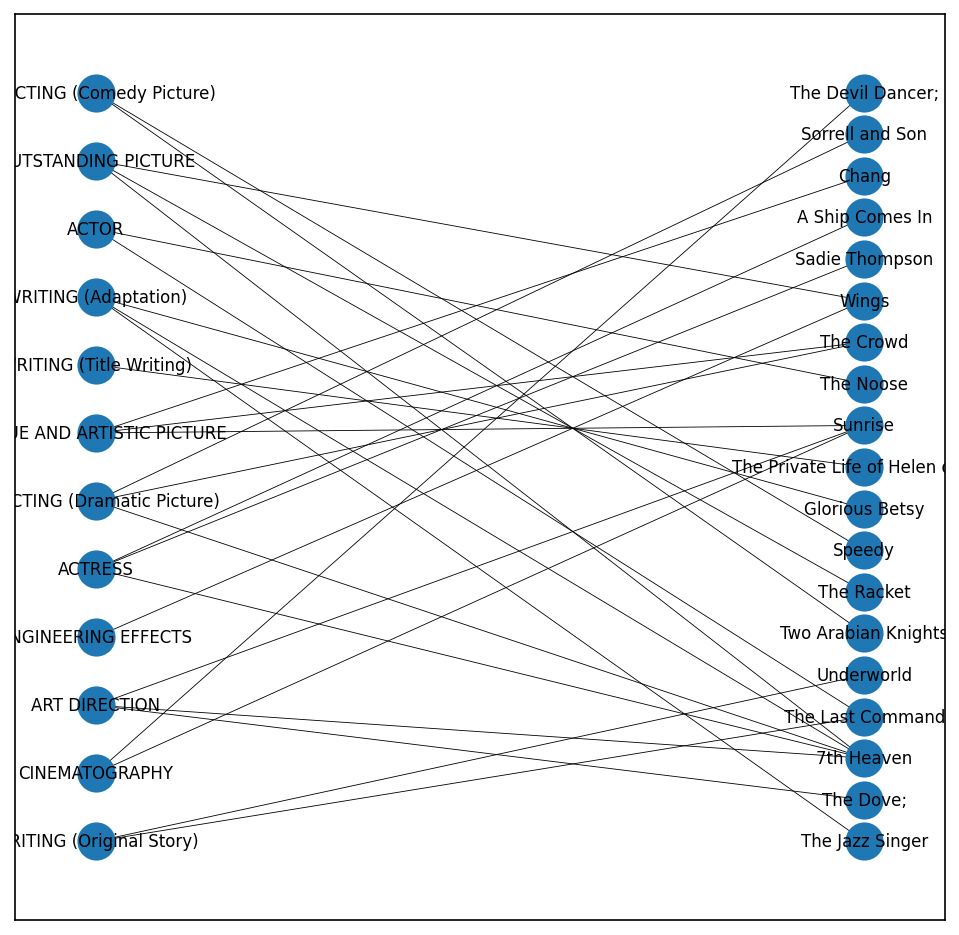
- Số phim là 19

- Số cạnh là 29

-Code hiển thị đồ thị 2 phía



*Hình 3. Code hiển thị đồ thị 2 phía*



*Hình 4. Đồ thị 2 phía*

- Nhìn vào đồ thị có thể thấy một bộ phim có thể được nhiều đề cử và một đề cử nhưng có nhiều phim đạt được.

- Đồ thị 1 phía

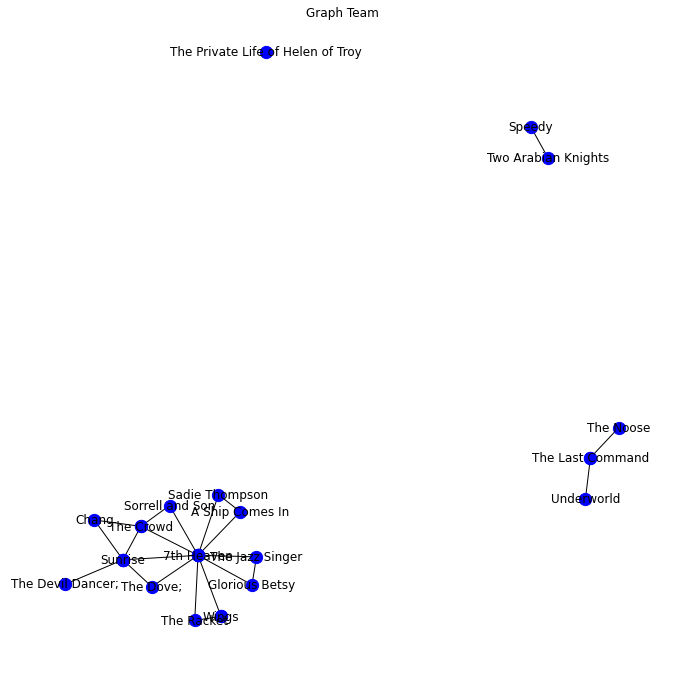
- Node: là bộ phim được đề cử (category).

- Edge: Hai bộ phim cùng được một đề cử sẽ được nối với nhau để tạo thành cạnh, ý nghĩa nói lên sự cạnh tranh giữa hai bộ phim để giành được đề cử đó trong giải Oscar.

- Weight : Trọng số là số đề cử trùng nhau mà hai bộ phim đều đạt được.

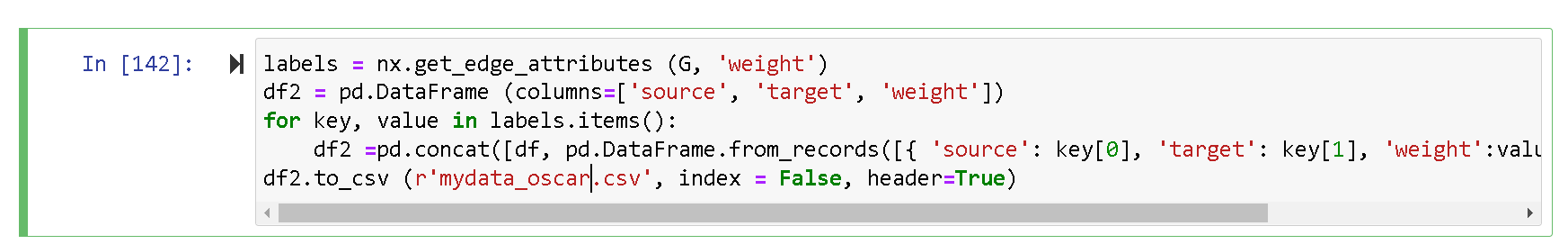
**

*Hình 5. Code hiển thị đồ thị 1 phía*

**

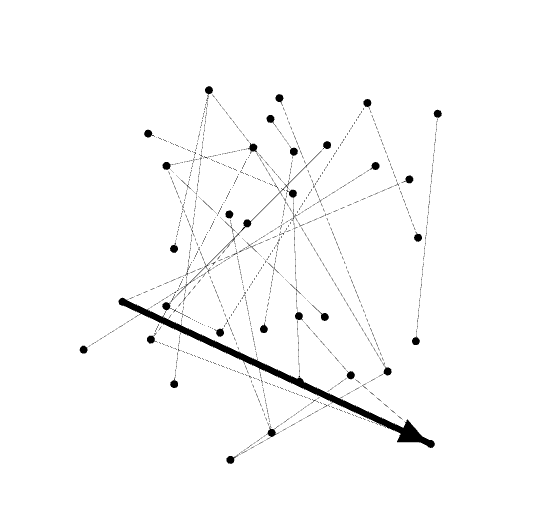
*Hình 6. Đồ thị 1 phía*

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

**

*Hình 7. Xuất dữ liệu đồ thị 1 phía ra file csv*

- Đồ thị 1 phía trên Gephi

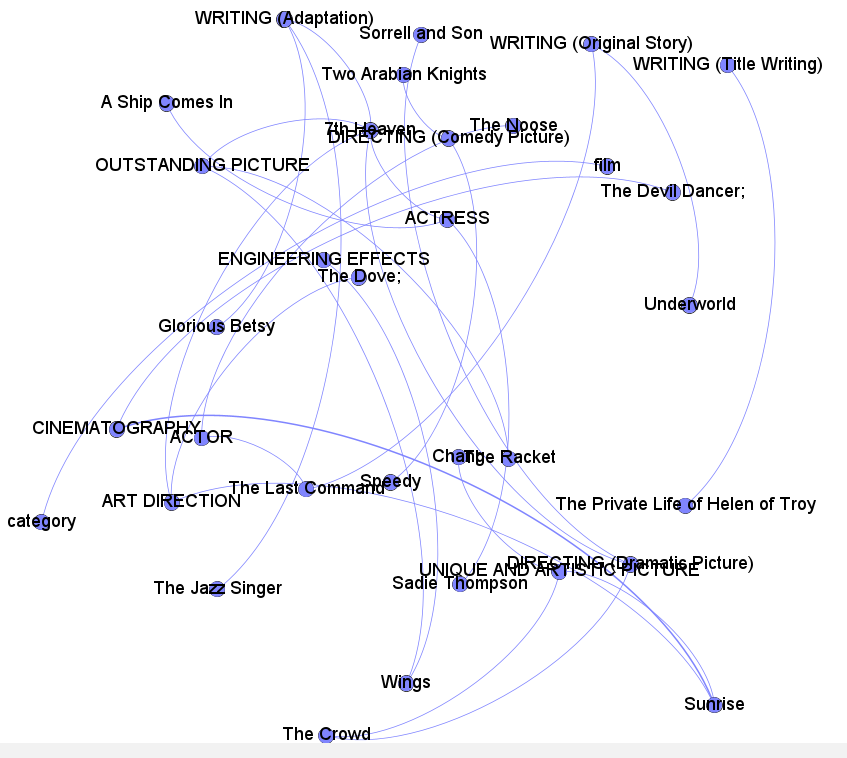
**

*Hình 8. Đồ thị một phía trên Gephi*

# **IV. THUẬT TOÁN**

## **1. Thuật toán phát hiện cộng đồng**

## **a. Thực thi thuật toán Louvain bằng Gephi**



*Hình 9. Đồ thị bằng Louvain*

Dễ thấy thuật toán rút ra được 5 cụm.

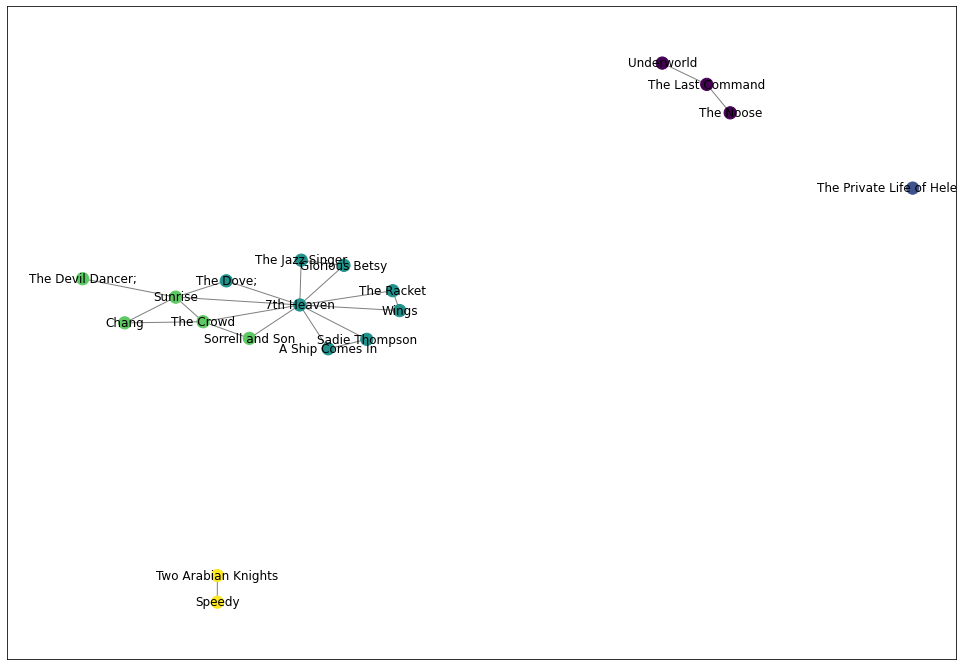
## **b. Thực thi bằng Python**

import matplotlib.cm as cm  
import matplotlib

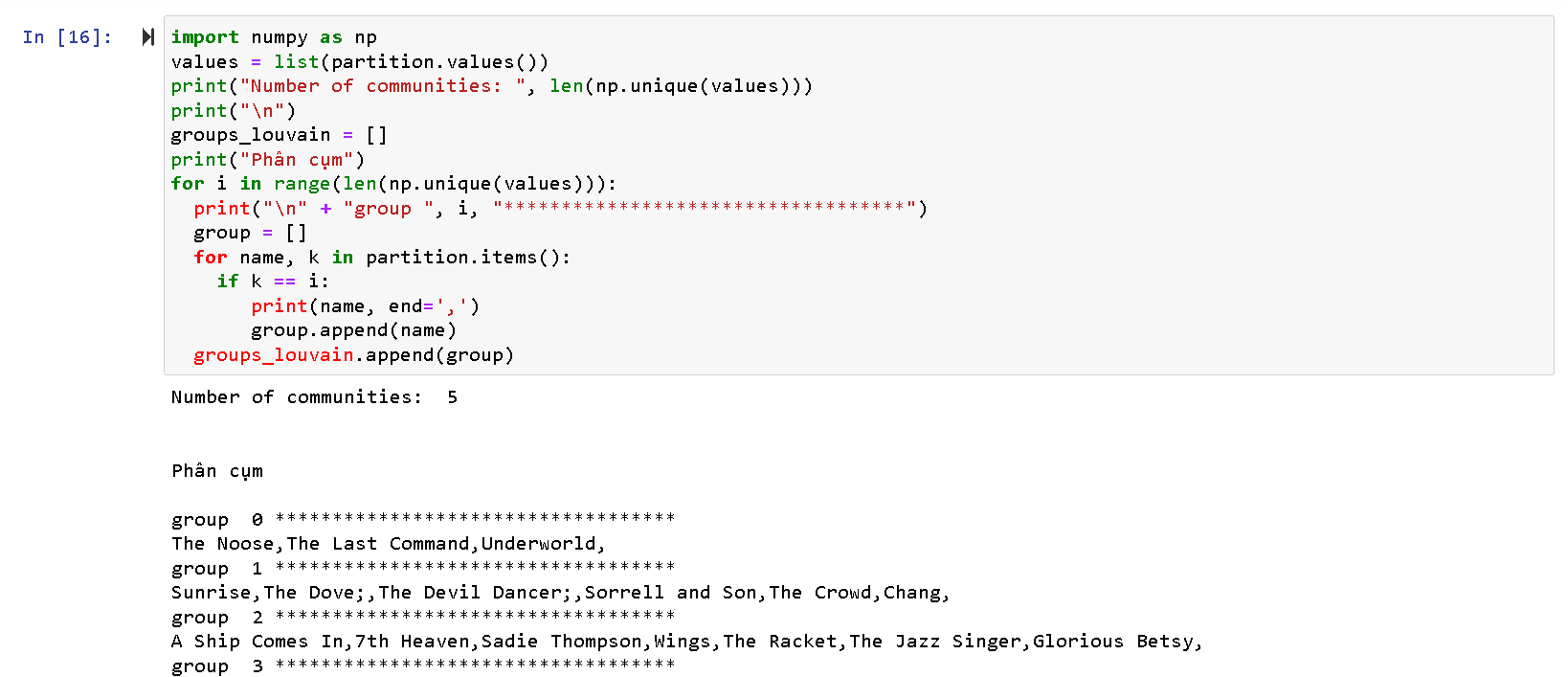
+ Thực thi thuật toán bằng Python:



*Hình 10. Code chạy thuật toán Louvain*



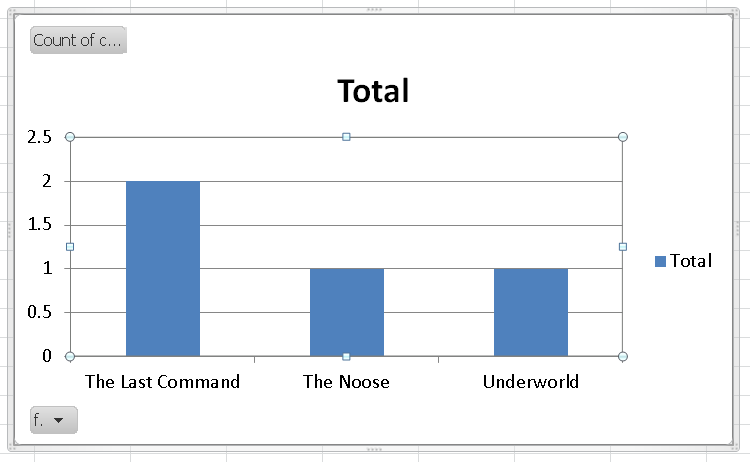
*Hình 11. Đồ thị phân cụm sử dụng Louvain*

**

*Hình 12. Danh sách các cụm*

Số cụm: 5

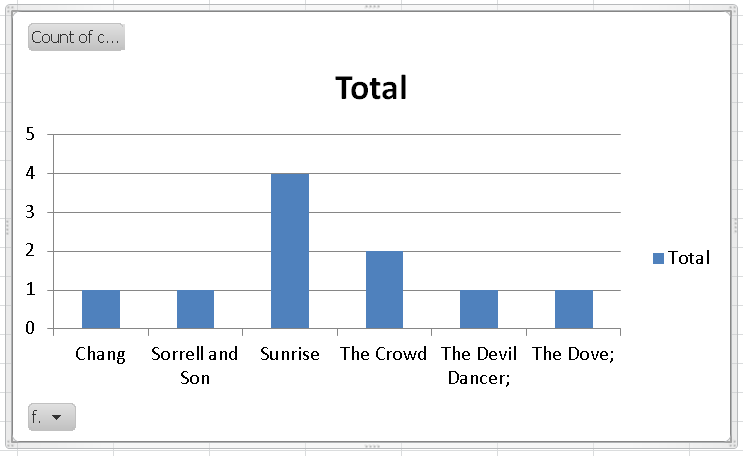
-Cụm thứ 0: The Noose, The Last Command, Underworld



*Hình 13. Biểu đồ cụm 0*

-Ý nghĩa cụm thứ 0: Những bộ phim cùng thể loại Original Story có cùng đề cử ACTOR, WRITING (Original Story)

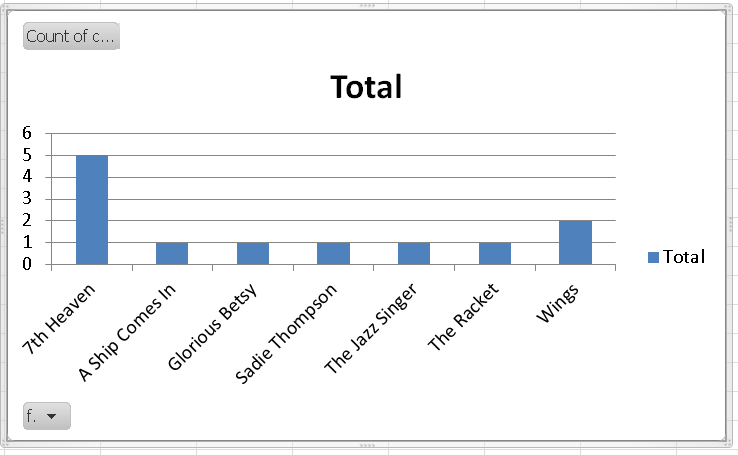
-Cụm thứ 1: Sunrise, The Dove, The Devil Dancer, Sorrell and Son, The Crowd, Chang



*Hình 14. Biểu đồ cụm 1*

-Ý nghĩa cụm thứ 1: Những bộ phim cùng thể loại phim lãng mạn có cùng đề cử ART DIRECTION, CINEMATOGRAPHY, UNIQUE AND ARTISTIC PICTURE

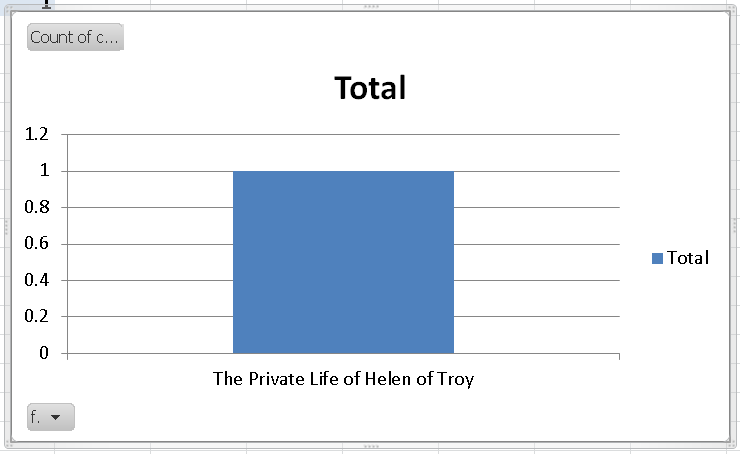
-Cụm thứ 2: A Ship Comes In, 7th Heaven, Sadie Thompson, Wings, The Racket, The Jazz Singer, Glorious Betsy



*Hình 15. Biểu đồ cụm 2*

-Ý nghĩa cụm thứ 2: Những bộ phim cùng thể loại storytelling có cùng đề cử ACTRESS, ART DIRECTION, DIRECTING (Dramatic Picture), OUTSTANDING PICTURE, WRITING (Adaptation)

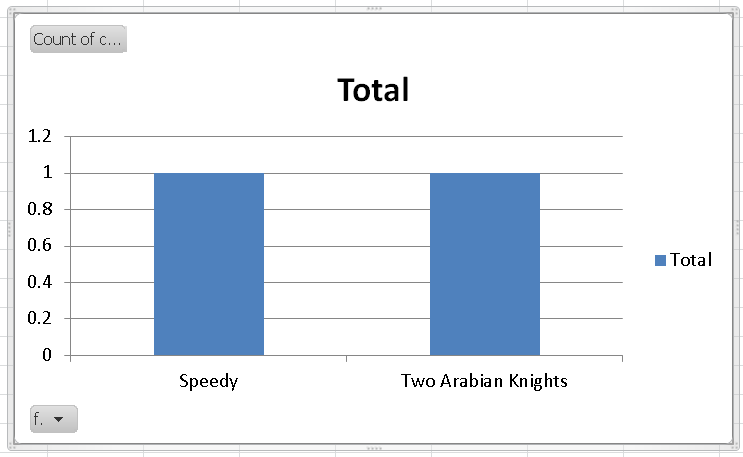
-Cụm thứ 3: The Private Life of Helen of Troy



*Hình 16. Biểu đồ cụm 3*

-Ý nghĩa cụm thứ 3: Những bộ phim cùng thể loại phim câm có cùng đề cử WRITING (Title Writing)

-Cụm thứ 4: Two Arabian Knights, Speedy



*Hình 17. Biểu đồ cụm 4*

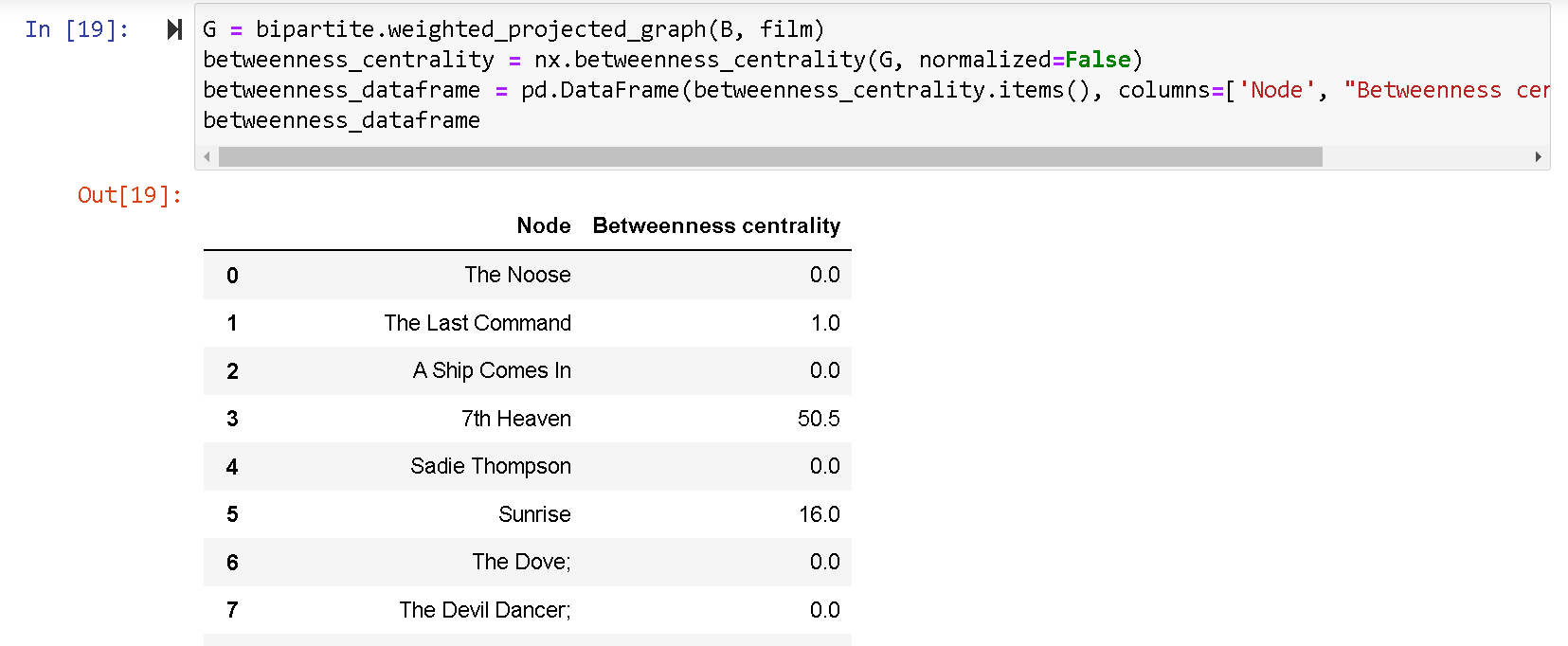
-Ý nghĩa cụm thứ 4: Những bộ phim cùng thể loại phim hài câm có cùng đề cử DIRECTING (Comedy Picture)

## **1. Centralities**

## **a. Thực thi thuật toán Betweenness Centrality**

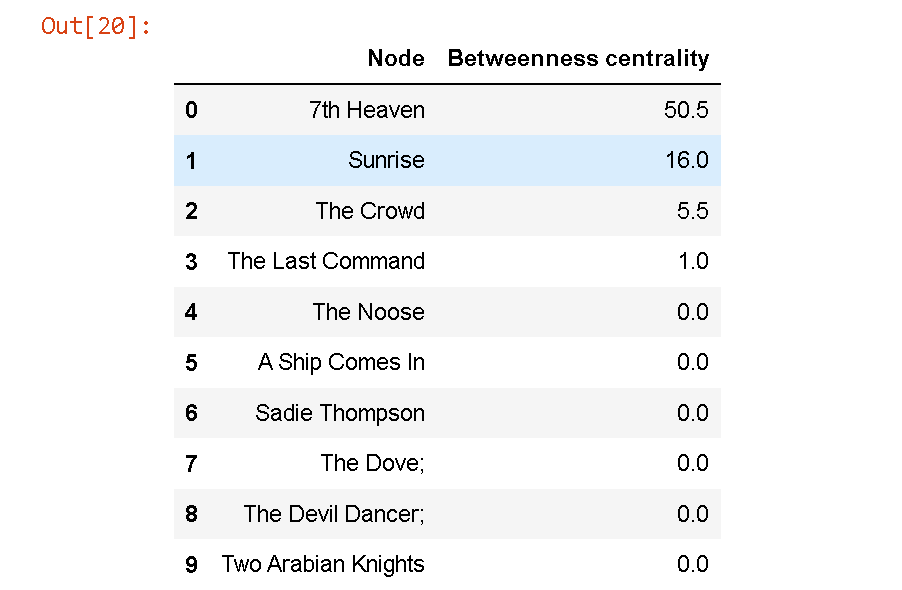
Kết quả của betweenness centrality thể hiện muốn đi từ node này tới node kia thì phải đi qua node trung tâm. Bộ phim nào có betweenness centrality càng cao thì phim đó đạt được nhiều hạng mục đề cử.

+ Thực thi thuật toán bằng Python:



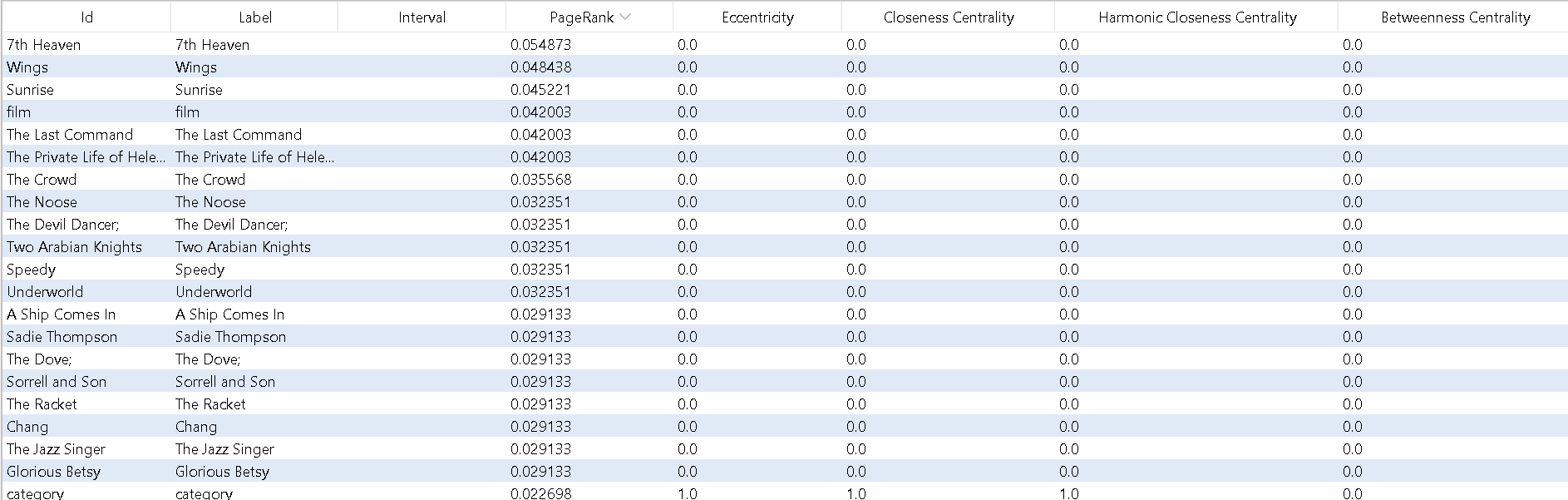
*Hình 18. Code và kết quả thuật toán Betweenness Centrality bằng python*

+ Top 10 phim có Betweenness Centrality cao nhất:



*Hình 19. Kết quả top 10 phim có Betweenness Centrality cao nhất*

+ Thực thi thuật toán bằng Gephi:



*Hình 20. Kết quả thuật toán Betweenness Centrality bằng Gephi*

## **Thực thi thuật toán Closeness Centrality**

Kết quả của closeness centrality thể hiện top độ gần các node đến tất cả các node trong mạng. Bộ phim nào có closeness centrality càng cao thì phim có các đề cử mà phim khác có.

+ Thực thi thuật toán bằng Python:



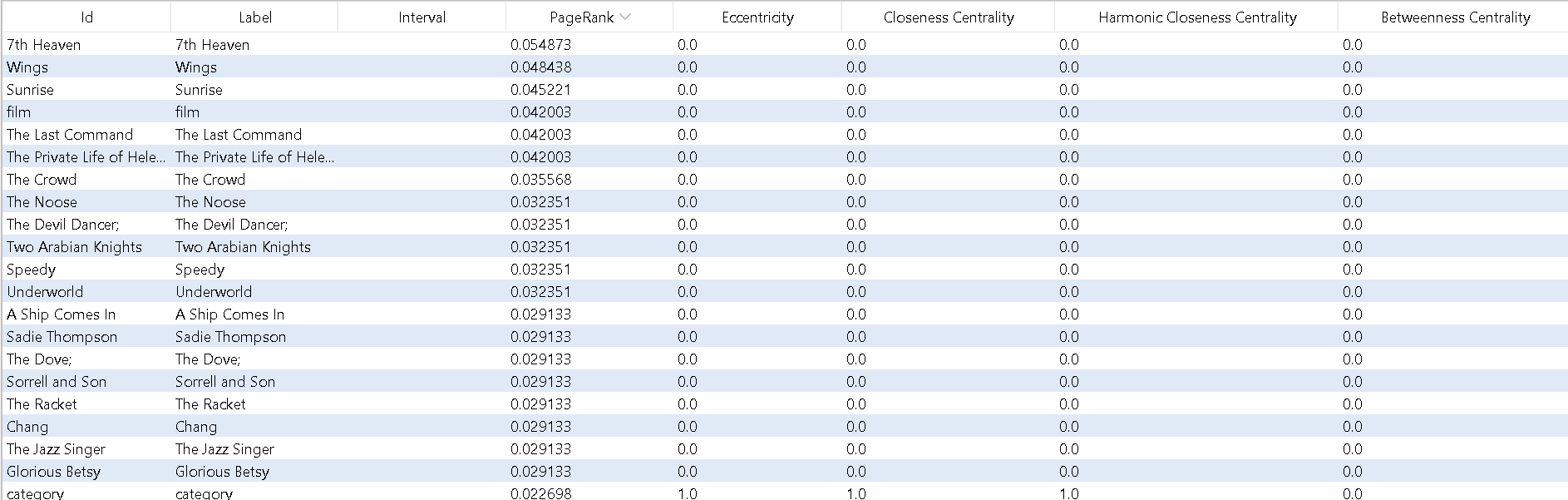
*Hình 21. Code và kết quả thuật toán Closeness Centrality bằng python*

+ Top 10 phim có Closeness Centrality cao nhất:



*Hình 22. Kết quả top 10 phim có Closeness Centrality cao nhất*

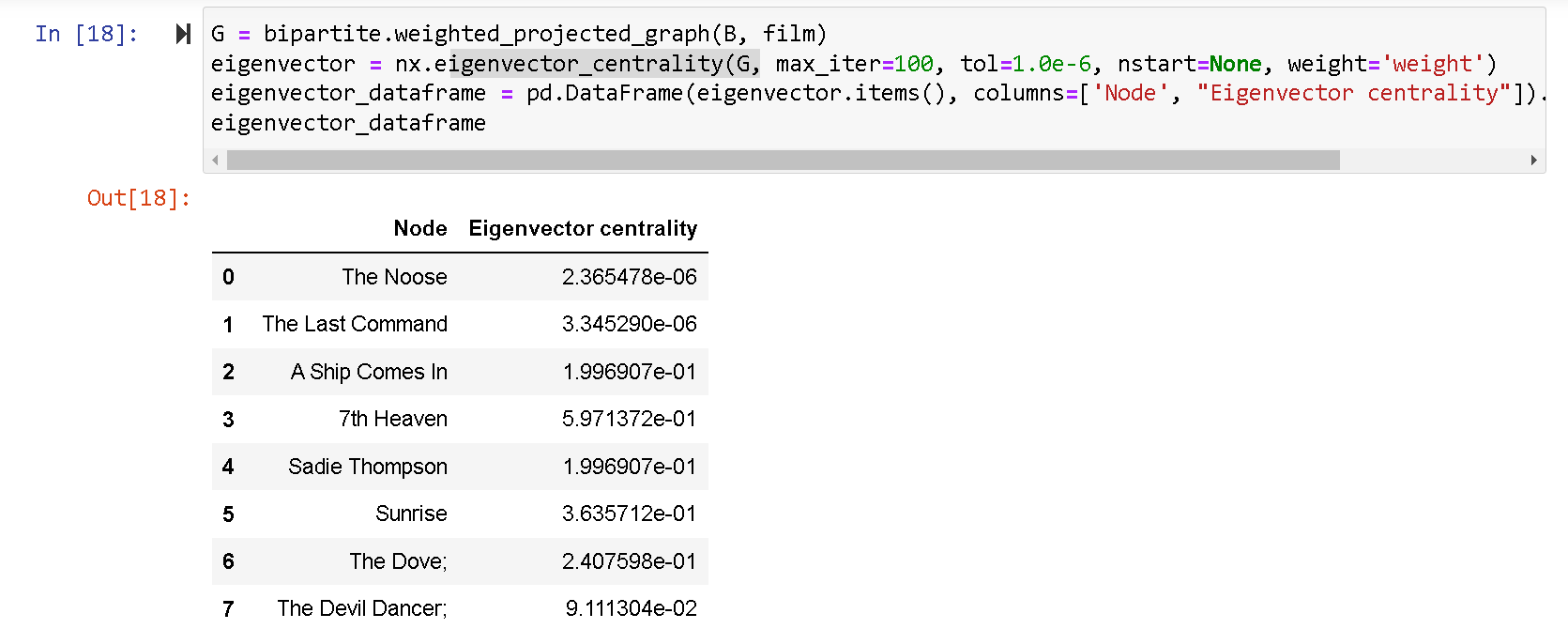
+ Thực thi thuật toán bằng Gephi:



*Hình 23. Kết quả thuật toán Closeness Centrality bằng Gephi*

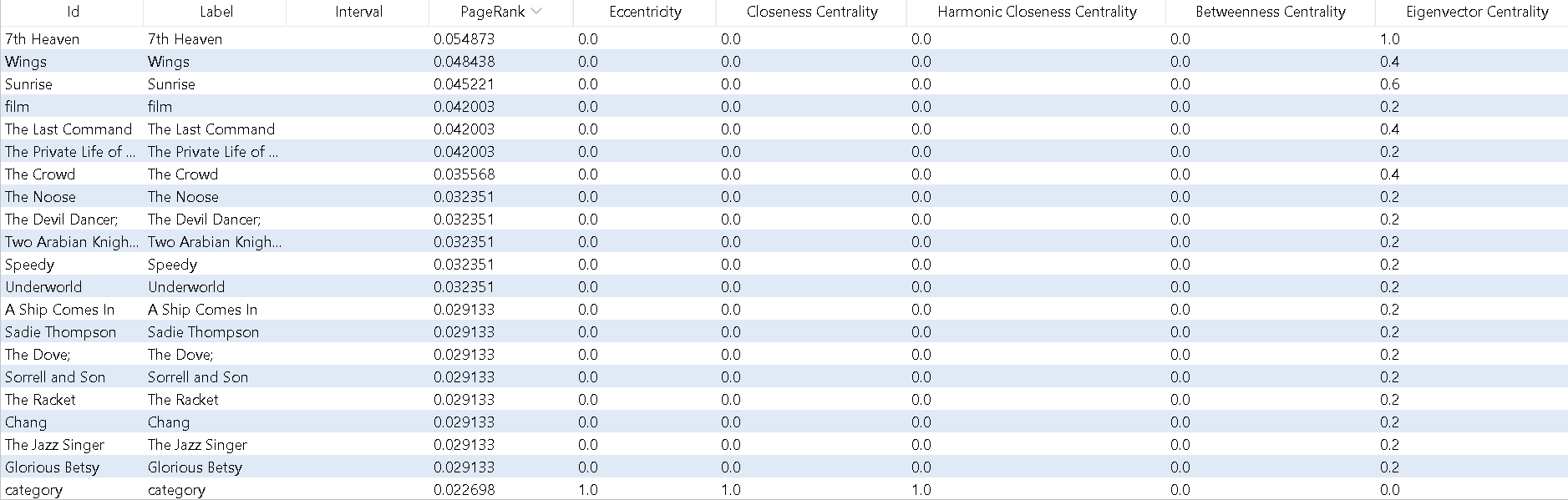
## **Thực thi thuật toán Eigenvector Centrality**

+ Thực thi thuật toán bằng Python:



*Hình 24. Code và kết quả thuật toán Eigenvector Centrality bằng python*

+ Thực thi thuật toán bằng Gephi:

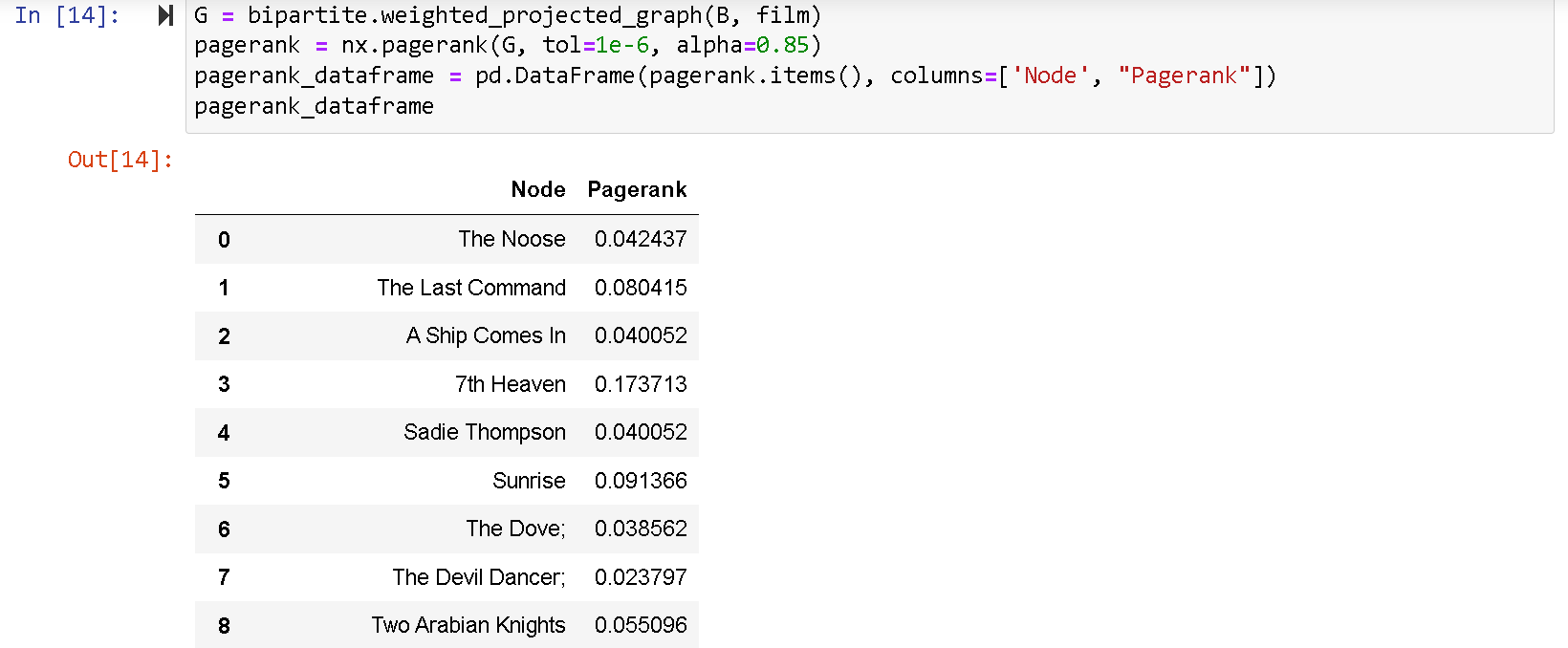


*Hình 25. Kết quả thuật toán Eigenvector Centrality bằng Gephi*

## **Thực thi thuật toán Page Rank**

Kết quả của page rank thể hiện bộ phim nào có xếp hạng càng cao thì phim đó càng được nhiều đề cử.

+ Thực thi thuật toán bằng Python:



*Hình 26. Code và kết quả thuật Page Rank bằng python*

+ Top 10 phim có Page Rank cao nhất:



*Hình 27. Kết quả top 10 phim có Page Rank cao nhất*

+ Thực thi thuật toán bằng Gephi:



*Hình 28. Kết quả thuật toán Page Rank bằng Gephi*

## **2. Graph mining**

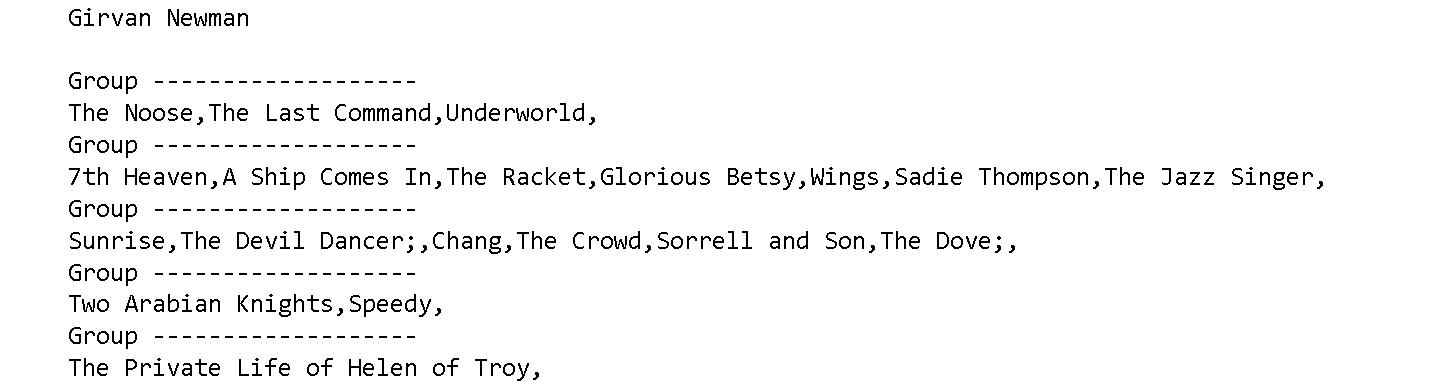
## **a. Thực thi thuật toán Girvan Newman**

+ Thực thi thuật toán bằng Python:

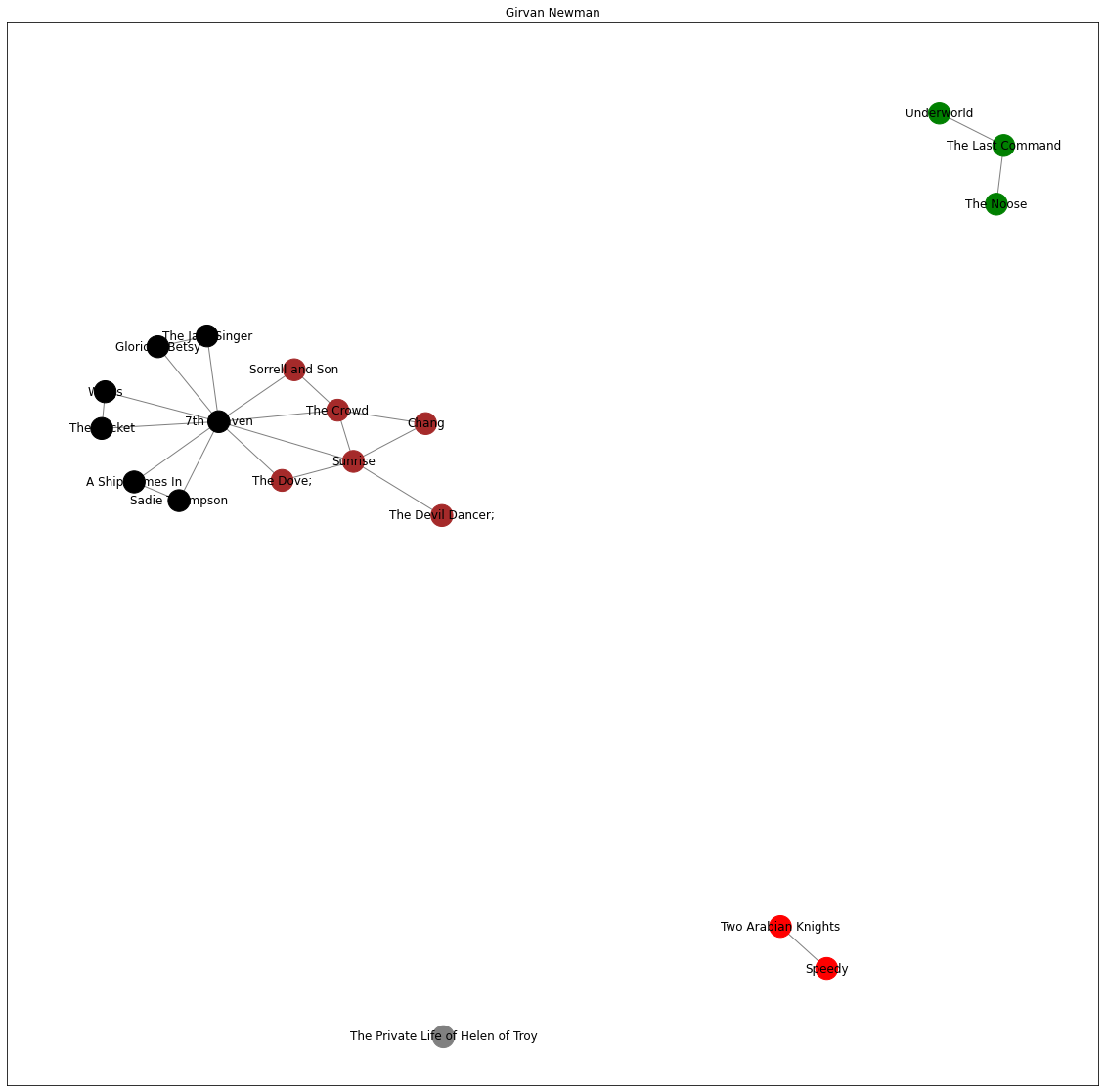


*Hình 29. Code thuật toán Girvan Newman bằng python*

+ Kết quả thuật toán Girvan Newman:



*Hình 30. Kết quả thuật toán Girvan Newman bằng python*

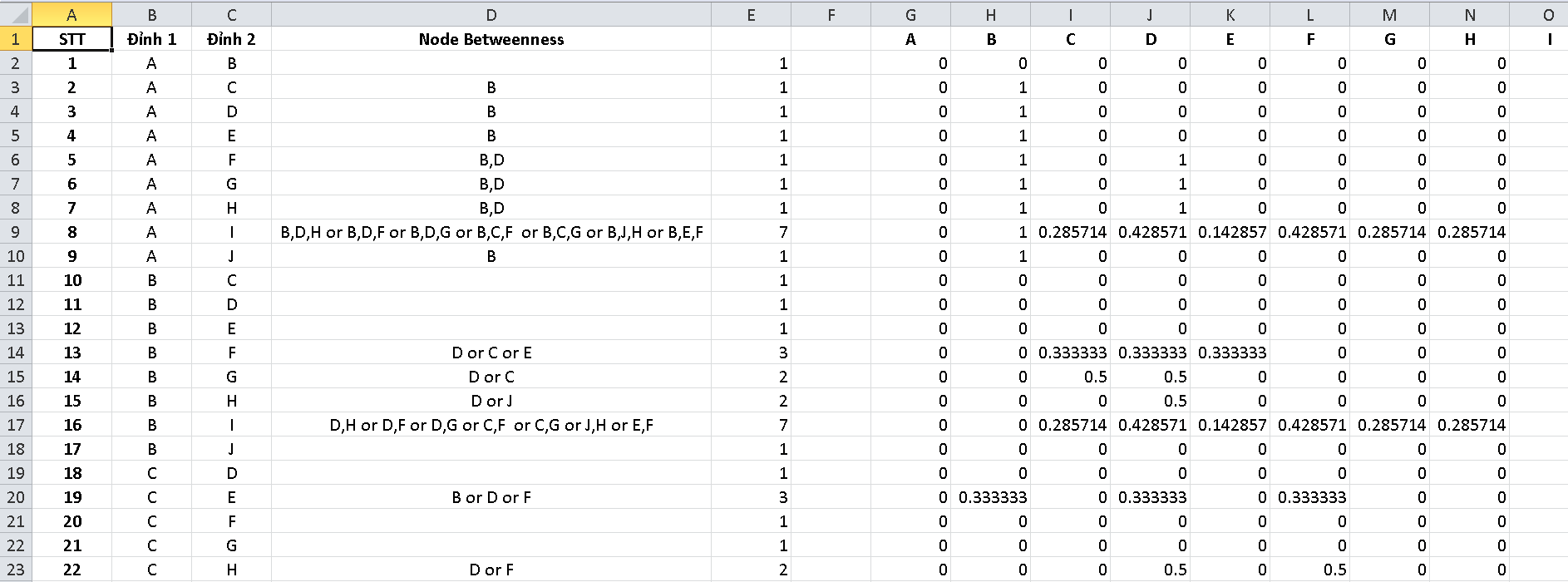


*Hình 31. Đồ thị thuật toán Girvan Newman*

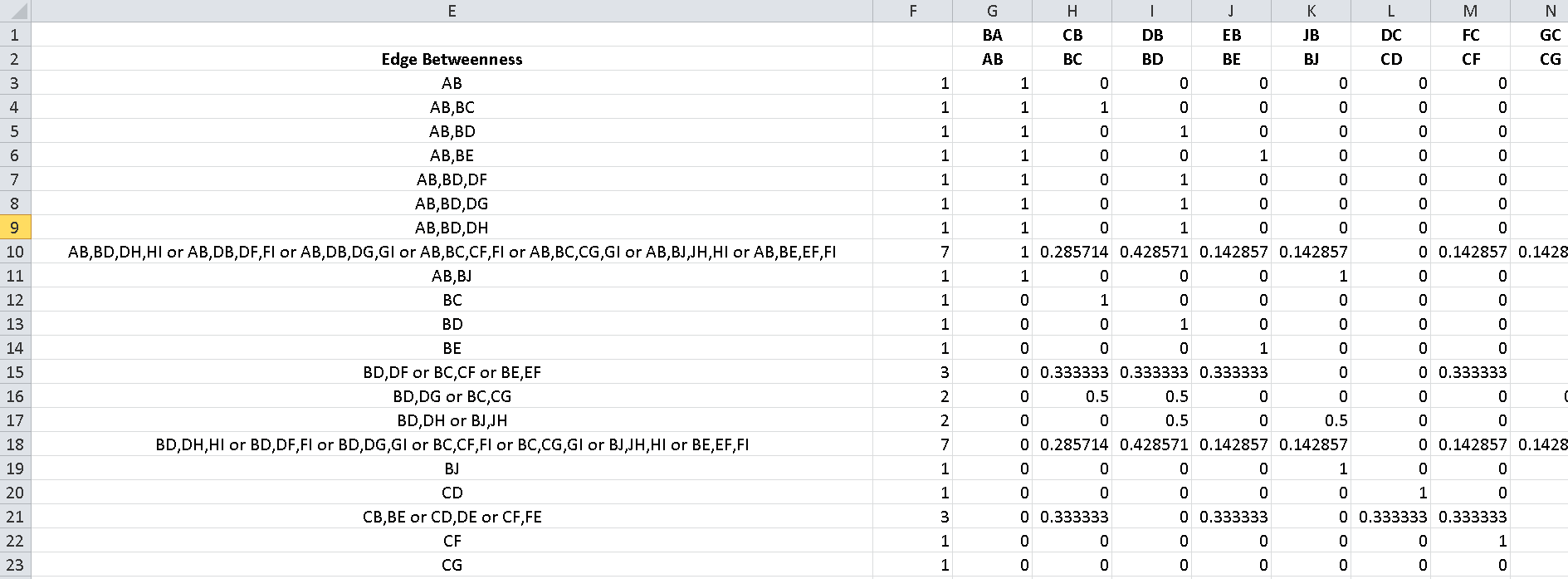
# **V. TÍNH TAY THUẬT TOÁN**

## **1. Centralities**

## **a. Thực thi thuật toán Betweenness Centrality**

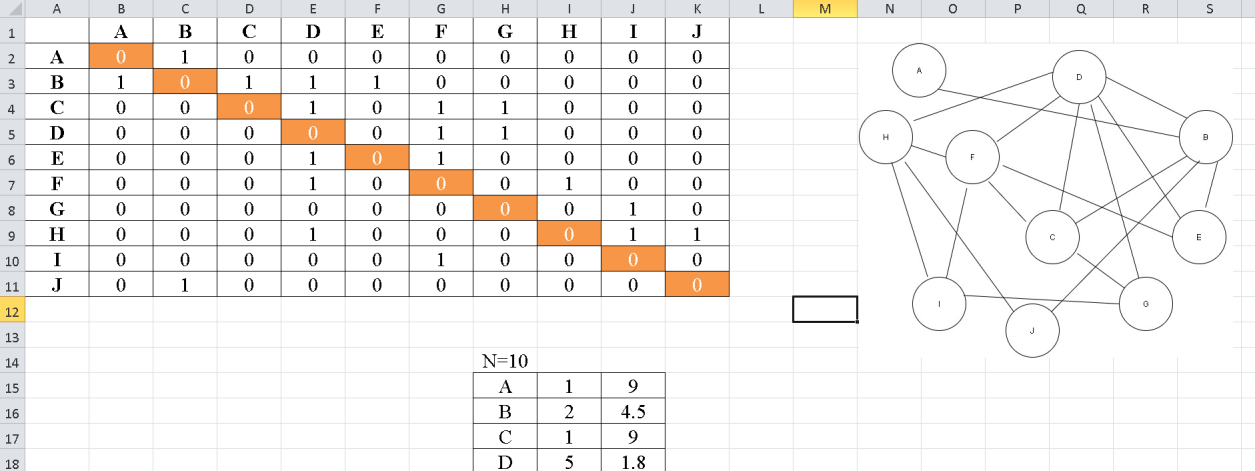


*Hình 32. Kết quả tính tay thuật toán Node Betweenness*



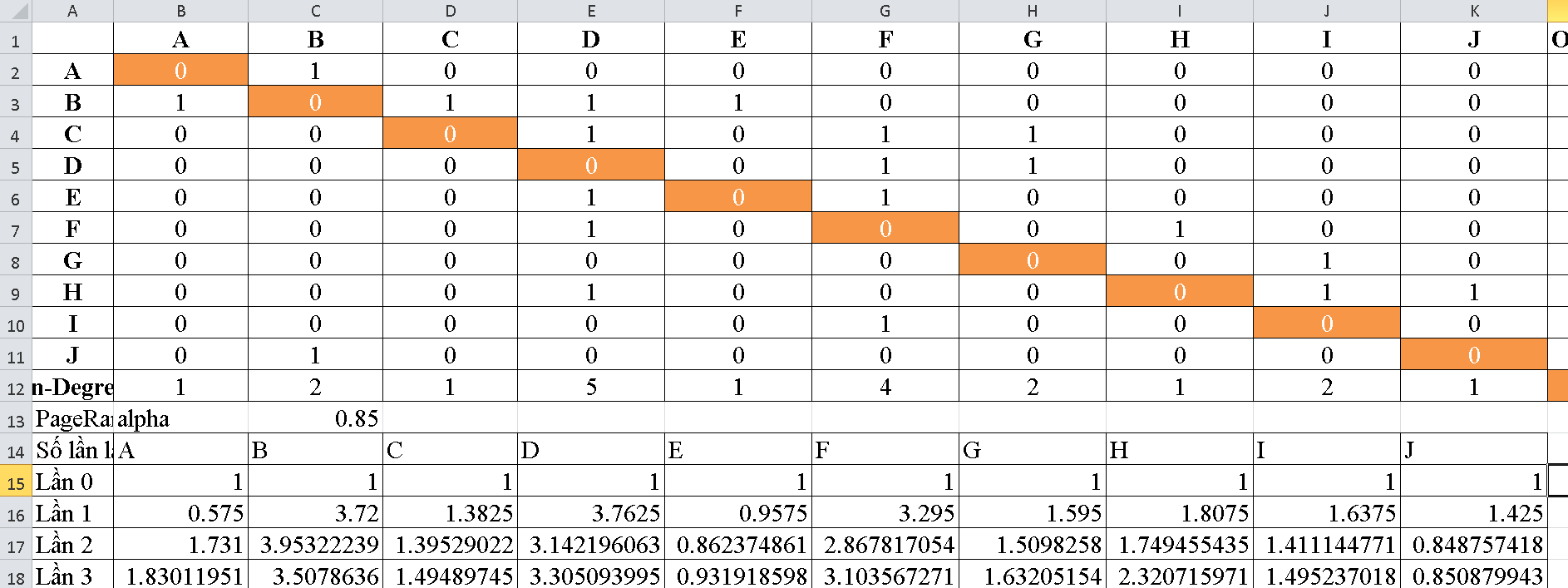
*Hình 33. Kết quả tính tay thuật toán Edge Betweenness*

## **Thực thi thuật toán Closeness Centrality**



*Hình 34. Kết quả tính tay thuật toán Closeness Centrality*

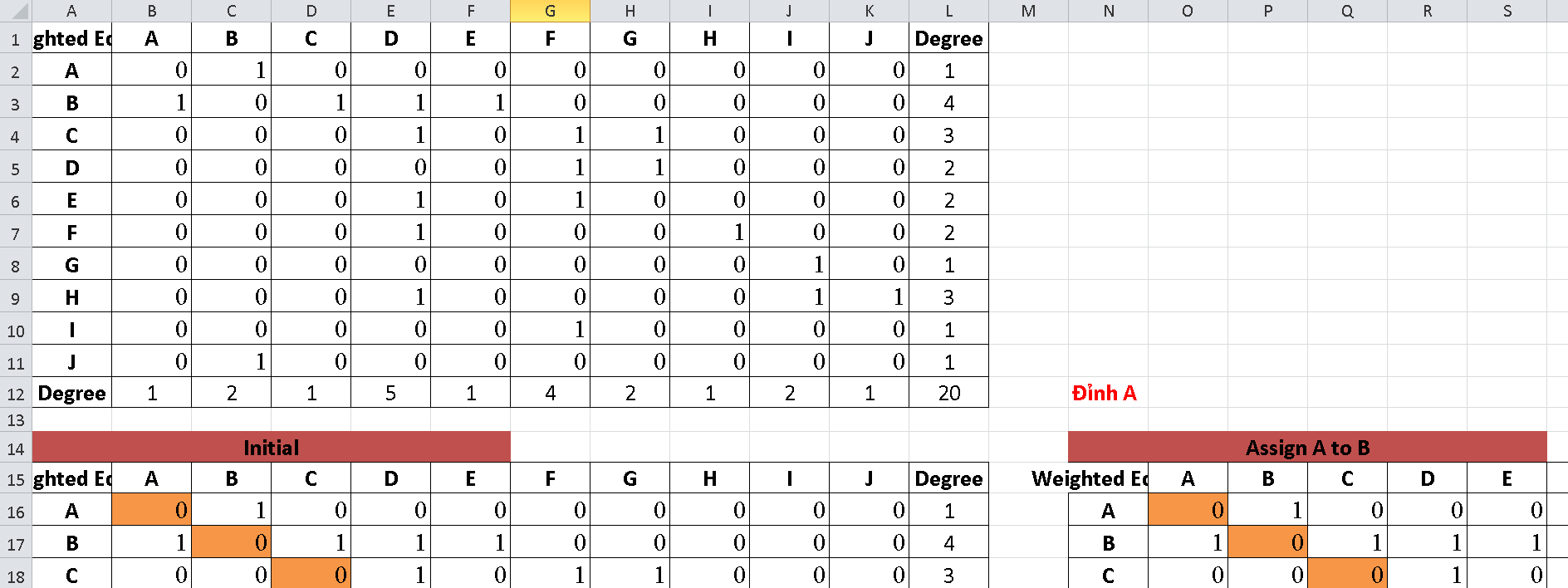
## **Thực thi thuật toán Page Rank**



*Hình 35. Kết quả tính tay thuật toán Page Rank*

## **2. Clustering**

## **a. Thực thi thuật toán Louvain**



*Hình 36. Kết quả tính tay thuật toán Louvain*

# **VI. TÀI LIỆU THAM KHẢO**

1. <https://drive.google.com/drive/folders/1C2nIZsyVODzIewcqtkleZ1BLt9g2vXDk>
2. <http://snap.stanford.edu/class/cs224w-2018/reports/CS224W-2018-3.pdf>
3. <https://www.geeksforgeeks.org/closeness-centrality-centrality-measure/>
4. <http://web.stanford.edu/class/cs224w/>
5. https://medium.com/analytics-vidhya/implement-louvain-community-detection-algorithm-using-python-and-gephi-with-visualization-871250fb2f25