**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯzỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC & KỸ THUẬT THÔNG TIN**

Logo, company name

Description automatically generated

**BÁO CÁO MÔN HỌC**

**MẠNG XÃ HỘI**

**IS353.N11.HTCL**

|  |  |
| --- | --- |
| **GIẢNG VIÊN HƯỚNG DẪN:** | **NGUYỄN THỊ KIM PHỤNG** |
| **SINH VIÊN THỰC HIỆN:** | **ĐÀO VĨNH KỲ - 19521730** |

**TP. HỒ CHÍ MINH, 12/2020**

Mục Lục

[Chương 1. GIỚI THIỆU iii](#_Toc120551521)

[1.1. Xác định bài toán iii](#_Toc120551522)

[Chương 2. DỮ LIỆU iii](#_Toc120551523)

[2.1. Giới thiệu nguồn dữ liệu iii](#_Toc120551524)

[2.2. Xử lý và phân tích dữ liệu iv](#_Toc120551525)

[2.2.1. Đồ thị 2 phía iv](#_Toc120551526)

[2.2.2. Đồ thị một phía vi](#_Toc120551527)

[2.2.3. Xuất đồ thị 1 phía ra file csv để thực hiện trên phần mềm Gephi viii](#_Toc120551528)

[2.2.4. Đồ Thị 1 phía trên Gephi ix](#_Toc120551529)

[Chương 3. THUẬT TOÁN PHÂN CỤM 10](#_Toc120551530)

[3.1. Thuật toán Louvain 10](#_Toc120551531)

[3.2. Thực thi thuật toán Louvain bằng Python 10](#_Toc120551532)

[Chương 4. THỰC THI THUẬT TOÁN XẾP HẠNG 13](#_Toc120551533)

[4.1. Thuật toán Pagerank (Thực thi bằng Gephi) 13](#_Toc120551534)

[4.2. Thuật toán Pagerank (Thực thi bằng Python) 13](#_Toc120551535)

[4.3. Betweenness Centrality 14](#_Toc120551536)

[4.4. Thuật toán Betweenness Centrality (Thực thi bằng Python) 14](#_Toc120551537)

[4.5. Thuật toán Betweenness Centrality (Thực thi bằng Gephi) 15](#_Toc120551538)

[4.6. Closeness Centrality 16](#_Toc120551539)

[4.7. Thuật toán Closeness Centrality (Thực thi bằng Python) 16](#_Toc120551540)

[4.8. Thuật toán Closeness Centrality (Thực thi bằng Gephi) 17](#_Toc120551541)

[Chương 5. MINI PROJECT (Tính tay 10 node ngẫu nhiên ) 17](#_Toc120551542)

# GIỚI THIỆU

* Đọc sách là cách tốt nhất để tiếp thu văn hóa thế giới, làm giàu thêm vốn hiểu biết của bản thân. Đọc sách giúp kích thích các dây thần kinh não bộ, giảm chứng mất trí nhớ, giữ cho bộ não hoạt động, tránh lão hóa. Nếu thực hiện việc đọc sách nhiều lần sẽ khiến chúng ta trở nên thông minh hơn.
* Ngày nay khối lượng sách đồ sộ, lượng kiến thức có trong quyển sách này lại có thể giống hệt những kiến thức trong quyển sách khác, chỉ khác nhau ở lớp bìa bên ngoài làm cho người đọc mất phương hướng, không biết nên lựa chọn thế nào cho thích hợp.
* Thế nên là qua môn học mạng xã hội ta có thể phân tích được các nhà xuất bản , các cuốn sách nổi tiếng được nhiều nhà xuất bản phát hành nhất bằng các thuật toán xếp hạng , phân cụm

## Xác định bài toán

Input : Nguồn dữ liệu sách mà hàng triệu người dùng cung cấp trong cộng đồng Book-Crossing

Output : Phân tích các cuốn sách được nhiều nhà xuất bản phát hành bằng các thuật toán xếp hạng , phân cụm.

# DỮ LIỆU

## Giới thiệu nguồn dữ liệu

* https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset
* Được thu thập bởi Cai-Nicolas Ziegler trong 4 tuần thu thập dữ liệu (tháng 8 / tháng 9 năm 2004) từ cộng đồng Book-Crossing với sự cho phép của Ron Hornbaker, CTO của Humankind Systems. Chứa 278.858 người dùng (ẩn danh nhưng có thông tin nhân khẩu học) cung cấp 1.149.780 xếp hạng (rõ ràng/ẩn ý) về 271.379 cuốn sách.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu dữ liệu | Mô tả |
| 1 | ISBN | Float | Mã số tiêu chuẩn quốc tế |
| 2 | Book-Title | char | Tên sách |
| 3 | Book-Author | char | Tác giả |
| 4 | Publisher | char | Nhà xuất bản |
| 5 | Year-Of-Publication | int | Năm xuất bản |

## Xử lý và phân tích dữ liệu

* Chuyển đổi dataframe thành đồ thị

### Đồ thị 2 phía

* Node: Là các nhà xuất bản và cuốn sách .
* Edge: Mối quan hệ giữa việc nhà xuất bản cùng xuất bản một cuốn sách

Text

Description automatically generated

* Chúng ta có thể thấy có 2038 cuốn sách, 608 nhà xuất bản và 2093 cạnh của đồ thị.
* Hiển thị đồ thị hai phía

Text

Description automatically generated

**A picture containing text, silhouette, screenshot

Description automatically generated**

Nhìn vào đồ thị có thể thấy mỗi nhà xuất bản có thể xuất bản rất nhiều quyển sách ngược lại một quyển sách cũng được rất nhiều nhà xuất bản cùng xuất bản.

### Đồ thị một phía

* Node: Là nhà xuất bản (Publisher)
* Edge: Các nhà xuất bản cùng xuất bản 1 cuốn sách sẽ hợp thành 1 cạnh
* Điều này nói lên rằng mỗi nhà xuất bản cùng xuất bản 1 cuốn sách sẽ cạnh tranh với nhau
* Weight: Trọng số là cuốn sách được các nhà xuất bản xuất bản trùng nhau

**Text

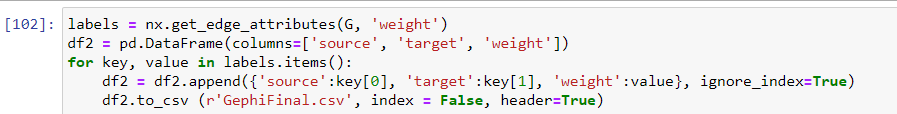
Description automatically generated**

*Code hiển thị đồ thị 1 phía*

**Diagram

Description automatically generated**

### Xuất đồ thị 1 phía ra file csv để thực hiện trên phần mềm Gephi

****

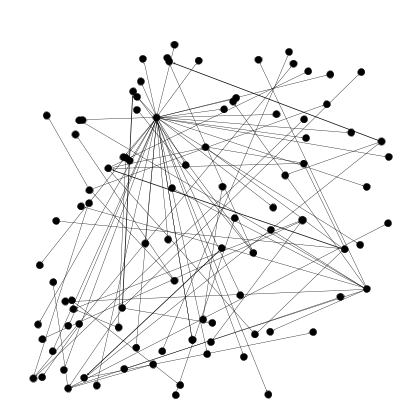
*Code xuất file csv để thực hiện trên Gephi*

* Source và Target là 2 node 2 đầu của cạnh, là 2 nhà xuất bản cùng xuất bản ít nhất một cuốn sách
* Weight: Trọng số, số nhà xuất bản chọn cùng một quyển sách để xuất bản

**Table

Description automatically generated**

### Đồ Thị 1 phía trên Gephi

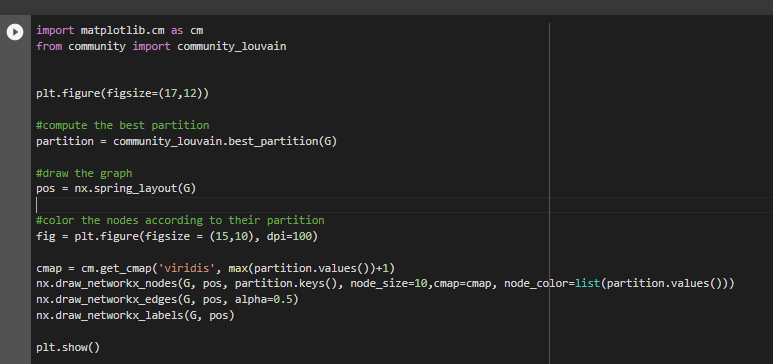
* **

# THUẬT TOÁN PHÂN CỤM

## Thuật toán 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.

## Thực thi thuật toán Louvain bằng Python



*Code chạy thuật toán Louvain*

* Đồ thị phân cụm sử dụng Louvain*

Text

Description automatically generated

*Danh sách các cụm*

* Số cụm 567
* Thuật toán Louvain không phù hợp để gom cụm với dataset này, bởi đặc tính riêng của từng cụm không thật sự nổi bật

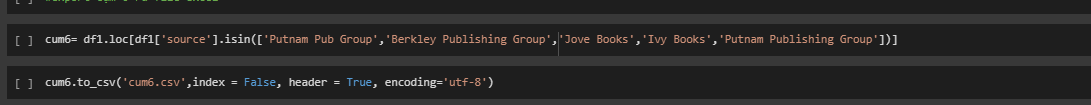
Text

Description automatically generated

Table

Description automatically generated

*Bảng chi tiết tất cả cụm trong Louvain*



*Xuất dữ liệu cụm 6 ra file excel*

*Biểu đồ cụm 6*

# THỰC THI THUẬT TOÁN XẾP HẠNG

## Thuật toán Pagerank (Thực thi bằng Gephi)

Ý nghĩa: PageRank có độ đo càng cao thì các nhà xuất bản đó càng có nhiều cuốn sách xuất bản chung với các nhà xuất bản khác. (Nghĩa là nhà xuất bản Random House ở đây là xuất bản chung với các nhà xuất bản khác cùng một cuốn sách cao nhất)

Graphical user interface, application

Description automatically generated

## Thuật toán Pagerank (Thực thi bằng Python)

A screenshot of a computer

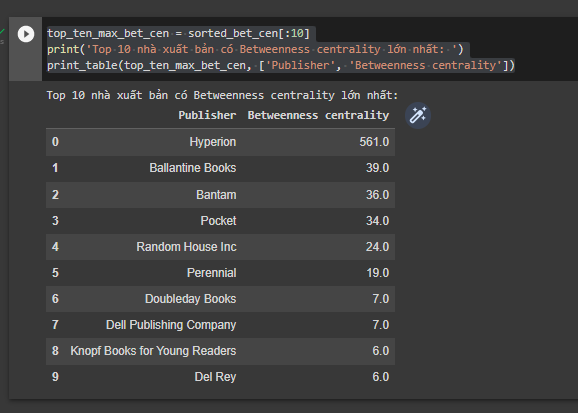
Description automatically generated with medium confidence

* Kết quả có không có sự khác biệt giữa thứ hạng của nhà xuất bản thông qua code Python và Gephi.

## 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.
* Ý nghĩa: Khi một người chọn mua sách để đọc , họ sẽ biết tới những cuốn sách khác thông qua các nhà xuất bản có betweeness cao

## Thuật toán Betweenness Centrality (Thực thi bằng Python)

**

## Thuật toán Betweenness Centrality (Thực thi bằng Gephi)

Table

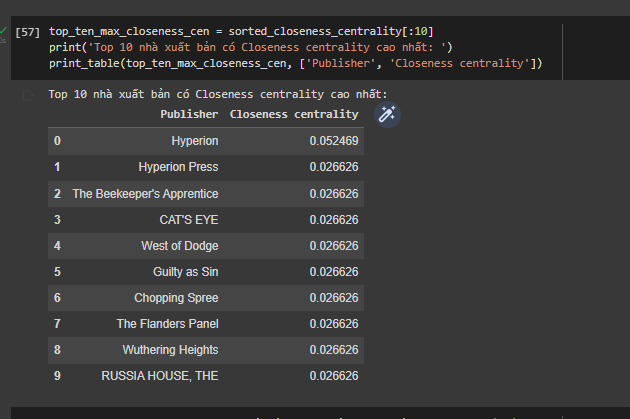
Description automatically generated

* Kết quả có không có sự khác biệt giữa thứ hạng của nhà xuất bản thông qua code python và Gephi

## Closeness Centrality

Kết quả của closeness centrality thể hiển top độ gần các node đến tất cả các node trong mạng .Ý nghĩa: Những nhà xuất bản có closeness centrality cao là những nhà xuất bản nhiều cuốn sách nhất trong nguồn dữ liệu ( có nghĩa là nhà xuất bản rất nổi tiếng , uy tín ) VD: Như ở Việt Nam thì NXB Kim Đồng là một nhà xuất bản rất nổi tiếng , do đó khi ta chọn sách mà nhà NXB Kim Đồng xuất bản thì độ tin cậy cao)

## Thuật toán Closeness Centrality (Thực thi bằng Python)



## Thuật toán Closeness Centrality (Thực thi bằng Gephi)

Table

Description automatically generated

# MINI PROJECT (Tính tay 10 node ngẫu nhiên )