ĐẶNG HOÀNG QUÂN **MSSV: 18520339**

MỤC LỤC

I. GIỚI THIỆU

Ngày nay, nhiều công ty phải đối mặt với một thách thức dường như đầy mâu thuẫn: làm thế nào để giảm chi phí vận hành đồng thời tăng mức độ dịch vụ khách hàng. Thiết kế mạng lưới chuỗi cung phù hợp cung cấp giải pháp cho cả hai vấn đề trên. Mặc dù có nhiều yếu tố cần xem xét khi thiết kế mạng lưới chuỗi cung ứng, nhưng quá trình này không quá phức tạp với các đối tác phù hợp.

Môn học mạng xã hội sẽ giúp phân tích và xem xét chuỗi cung ứng của công ty bằng cách sử dụng các thước đo về trọng tâm và phân cụm liên quan đến các khía cạnh nổi bật (dựa trên đường đi ngắn nhất, phổ, khoảng cách, ...) để từ đó đưa ra các chiến lược vận hành phù hợp nhất.

II. XÁC ĐỊNH BÀI TOÁN

- Input: Tập dữ liệu ban đầu trên nguồn dữ liệu Kaggle được qua tiền xử lý dữ liệu.
- Output: Đưa ra độ đo, cộng đồng phục vụ cho việc phân tích mạng xã hội **DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS.**

III. DỮ LIỆU

1. Giới thiệu nguồn dữ liệu
   - Link dataset: https://www.kaggle.com/shashwatwork/dataco-smart-supply-chain-for-big-data-analysis.
   - Dữ liệu **DataCo SMART SUPPLY CHAIN FOR BIG DATA ANALYSIS** được cung cấp trên nền tảng Kaggle chứa dữ liệu của chuỗi cung ứng được sử dụng bởi công ty DataCo Global, bao gồm tập hợp các sản phẩm đã bán của họ, chi tiết tài chính (lãi, lỗ, tổng doanh thu, v.v.), chi tiết giao hàng và chi tiết khách hàng như doanh số, nhân khẩu học và chi tiết giao dịch.
   - Dữ liệu có kích thước 91 MB bao gồm 180519 dòng với 54 thuộc tính:

Thuộc tính	Type	Mô tả	Các giá trị
Type	char	Loại giao dịch thực hiện	• CASH • DEBIT • PAYMENT • TRANSFER
Days for shipping (real)	int	Số ngày giao hàng thực tế	
Days for shipment (scheduled)	int	Số ngày giao hàng theo dự kiến	
Benefit per order	float	Thu nhập cho mỗi đơn hàng được đặt	= Order Profit Per Order
Sales per customer	float	Tổng doanh số theo khách hàng	= Order Item Total
Delivery Status	char	Trạng thái giao hàng của đơn hàng	• Advance shipping • Late delivery • Shipping canceled • Shipping on time
Late_delivery_risk	int	Biến phân loại cho biết gửi muộn hay không	• 1 – Gửi muộn • 0 – Không gửi muộn
Category Id	int	Mã danh mục sản phẩm	= Product Category Id
Category Name	char	Tên danh mục sản phẩm	
Customer City	char	Thành phố nơi khách hàng thực hiện mua hàng	
Customer Country	char	Đất nước nơi khách hàng thực hiện mua hàng	
Customer Email	char	Email của khách hàng	XXXXXXXXX

Customer Fname	char	Tên khách hàng	
Customer Id	int	ID khách hàng	
Customer Lname	char	Họ khách hàng	
Customer Password	char	Mật khẩu khách hàng	XXXXXXXXX
Customer Segment	char	Phân khúc khách hàng	• Consumer • Corporate • Home Office
Customer State	char	Tiểu bang của cửa hàng đã đăng ký giao dịch mua	
Customer Street	char	Đường của cửa hàng đã đăng ký giao dịch mua	
Customer Zipcode	float	Mã bưu điện khách hàng	
Department Id	int	Mã bộ phận của cửa hàng	
Department Name	char	Tên bộ phận của cửa hàng	
Latitude	float	Vĩ độ tương ứng với vị trí của cửa hàng	
Longitude	float	Kinh độ tương ứng với vị trí của cửa hàng	
Market	char	Thị trường của nơi được giao hàng	• Africa • Europe • LATAM • Pacific Asia • USCA

Order City	char	Thành phố của đơn hàng được đặt	
Order Country	char	Quốc gia của đơn hàng được đặt	
Order Customer Id	int	Mã đặt hàng của khách	= Customer Id
order date (DateOrders)	datetime	Ngày đặt hàng	
Order Id	int	Mã đơn hàng	
Order Item Cardprod Id	int	Mã sản phẩm được tạo thông qua đầu đọc RFID	= Product Card Id
Order Item Discount	float	Giá trị chiết khấu của mặt hàng trong đơn hàng	= Sales - Order Item Total
Order Item Discount Rate	float	Phần trăm chiết khấu của mặt hàng trong đơn hàng	
Order Item Id	int	Mã mặt hàng được đặt trong đơn hàng	
Order Item Product Price	float	Giá của sản phẩm khi không giảm giá	= Product Price
Order Item Profit Ratio	float	Tỷ lệ lợi nhuận của mặt hàng trong đơn hàng	
Order Item Quantity	int	Số lượng sản phẩm c mỗi mặt hàng trong đơn hàng	
Sales	float	Doanh số bán hàng	= Order Item Product Price * Order Item Quantity
Order Item Total	float	Tổng số tiền cho mặt hàng trong đơn hàng	= Sales per customer

Order Profit Per Order	float	Lợi nhuận cho mỗi mặt hàng trong đơn hàng	= Benefit per order
Order Region	char	Khu vực nơi đơn đặt hàng được tiến hành giao	• Southeast Asia • South Asia • Oceania • Eastern Asia • West Asia • West of USA • US Center • West Africa • Central Africa • North Africa • Western Europe • Northern • Caribbean • South America • East Africa • Southern Europe • East of USA • Canada • Southern Africa • Central Asia • Europe • Central America • Eastern Europe • South of USA
Order State	char	Tiểu bang nơi đơn hàng được tiến hành giao	

Order Status	char	Trạng thái đơn hàng	• CANCELED • CLOSED • COMPLETE • ON_HOLD • PAYMENT_REVIEW • PENDING • PENDING_PAYMENT • PROCESSING • SUSPECTED_FRAUD
Order Zipcode	float	Mã bưu điện đơn hàng	
Product Card Id	int	Mã sản phẩm	= Order Item Cardprod Id
Product Category Id	int	Mã danh mục sản phẩm	= Category Id
Product Description	float	Mô tả sản phẩm	
Product Image	char	Liên kết đến hình ảnh của sản phẩm	
Product Name	char	Tên sản phẩm	
Product Price	float	Giá sản phẩm	= Order Item Product Price
Product Status	int	Trạng thái sản phẩm	• 1 – Không có sẵn • 0 – Có sẵn
shipping date (DateOrders)	datetime	Ngày và thời gian chính xác của lô hàng	
Shipping Mode	char	Chế độ vận chuyển	• First Class • Same Day • Second Class • Standard Class

2. Xử lý và phân tích dữ liệu

2.1. Đọc dữ liệu từ file vào DataFrame

```
import matplotlib.pyplot as plt
plt.style.use('seaborn')
```

```
import pandas as pd
PRIMARY = 'Category Name'
SECONDARY = 'Order Region'

df = pd.read_csv(
    'DataCoSupplyChainDataset.csv',
    usecols = [PRIMARY, SECONDARY],
    encoding = 'unicode_escape'
).apply(lambda col: col.str.strip())
df.head()
```

	Category Name	Order Region
0	Sporting Goods	Southeast Asia
1	Sporting Goods	South Asia
2	Sporting Goods	South Asia
3	Sporting Goods	Oceania
4	Sporting Goods	Oceania

Hình 1. Đọc dữ liệu từ file vào DataFrame

2.2. Làm sạch dữ liệu

- Kiểm tra các giá trị bị khuyết ➔ không phát hiện giá trị nào nên không cần loại bỏ:

```
df.isnull().sum().sort_values(ascending=False)
```

```
Category Name    0
Order Region     0
```

Hình 2. Đọc dữ liệu từ file vào DataFrame

- Loại bỏ các giá trị trùng lặp ➔ Kết quả cuối cùng nhận được là 1 bộ dữ liệu gồm 1751 dòng và 2 cột:

```
df.drop_duplicates(inplace=True)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 691 entries, 0 to 162002
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Category Name  691 non-null    object
 1   Order Region   691 non-null    object
dtypes: object(2)
memory usage: 16.2+ KB
```

Hình 3. Loại bỏ các giá trị trùng lặp

2.3. Chuyển đổi DataFrame thành đồ thị

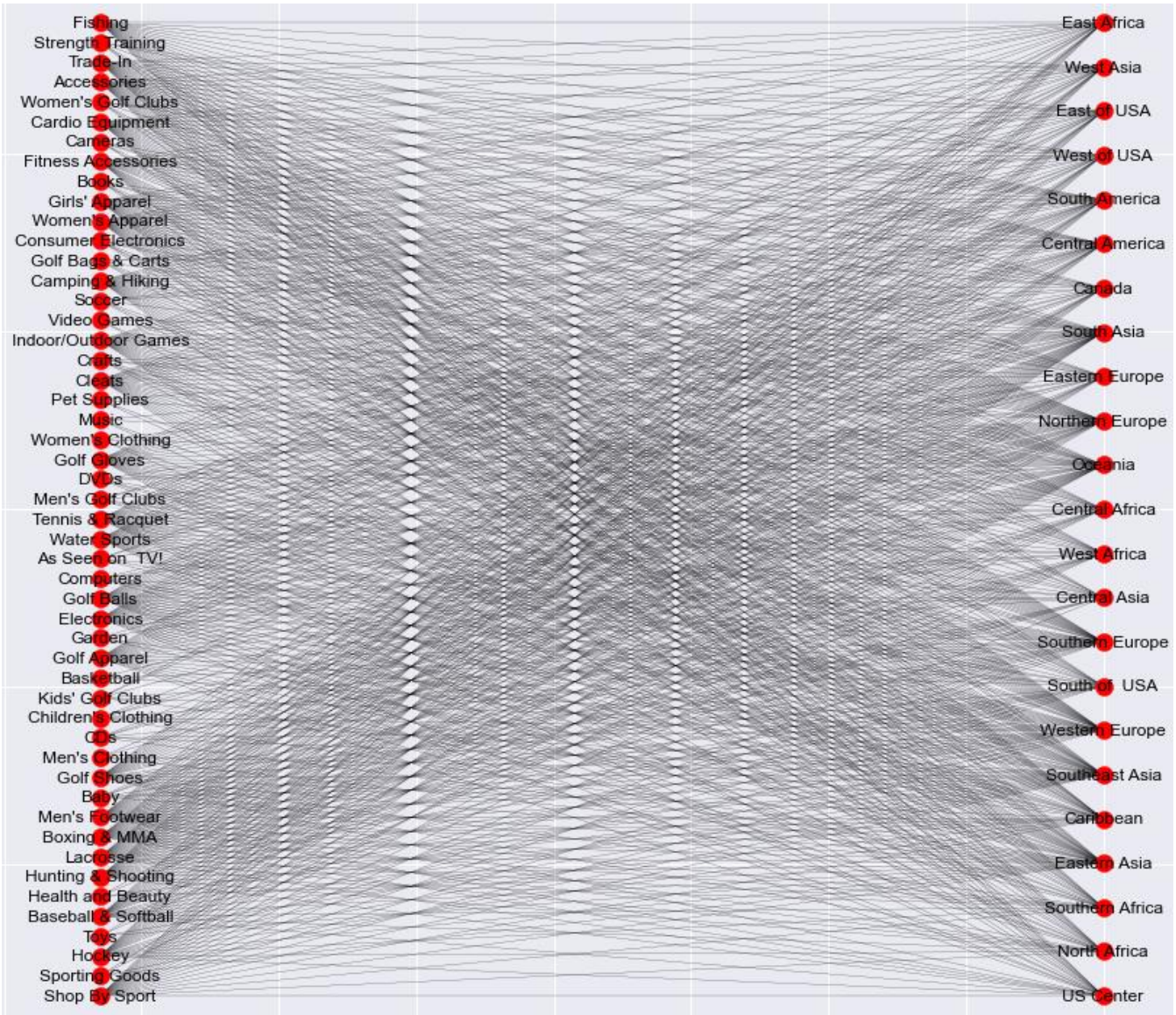
```
primary_col, secondary_col = df[PRIMARY], df[SECONDARY]
print(f'Number of {PRIMARY}:', primary_col.nunique())
print(f'Number of {SECONDARY}:', secondary_col.nunique())
print('Number of edges:', len(df))
```

```
Number of Category Name: 50
Number of Order Region: 23
Number of edges: 691
```

Hình 4. Chuyển đổi DataFrame thành đồ thị

2.2.1. Đồ thị 2 phía

- Node: danh mục các sản phẩm (Category Name) và khu vực nơi đơn đặt hàng được tiến hành giao (Order Region).
- Edge: mối quan hệ hình thành khi 1 loại sản phẩm được tiến hành giao từ 1 khu vực.



Hình 5. Đồ thị 2 phía

⇨ Nhìn vào đồ thị có thể thấy cùng 1 loại sản phẩm có thể được tiến hành giao từ nhiều khu vực và 1 khu vực cũng có thể tiến hành giao hàng nhiều loại sản phẩm.

- Code hiển thị đồ thị 2 phía:

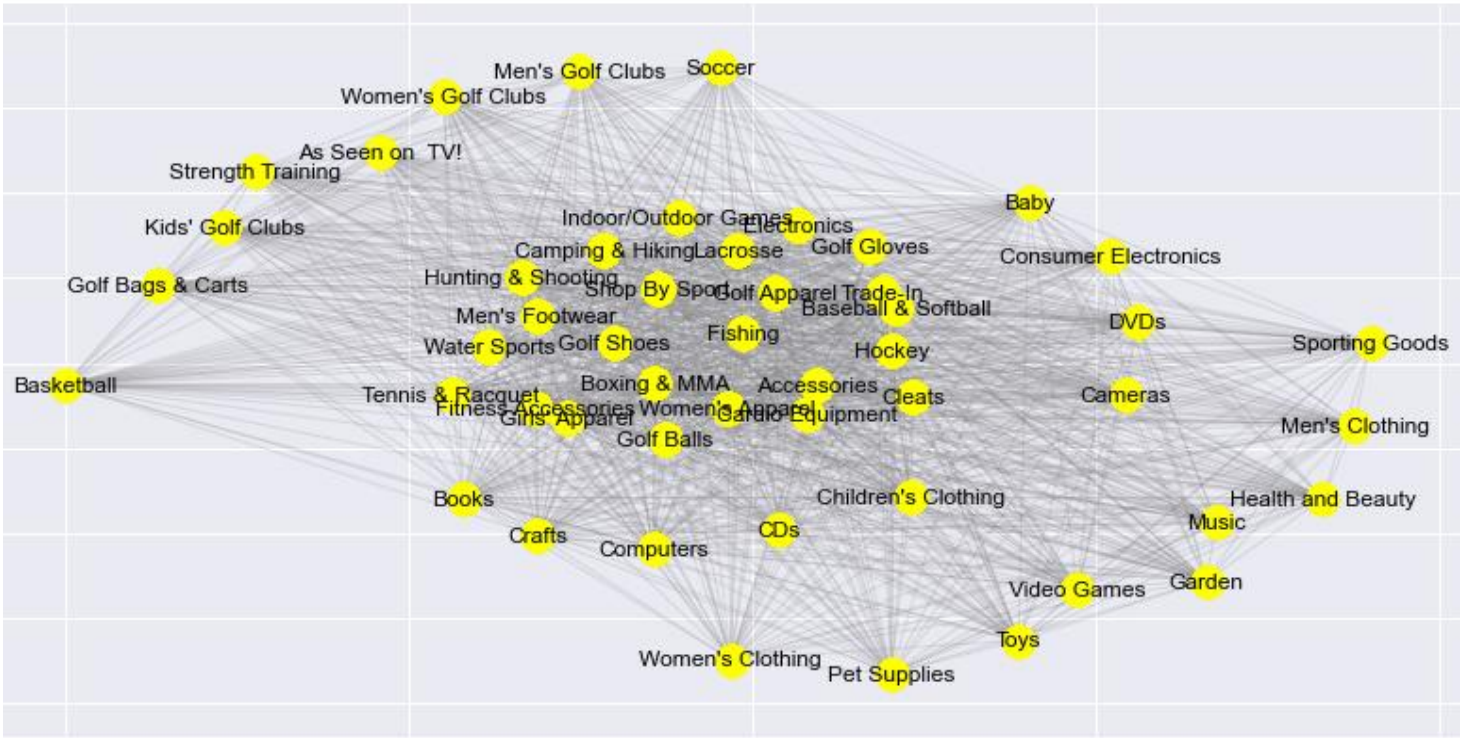
```
import networkx as nx
from networkx.algorithms import bipartite

B = nx.Graph()
for index, row in df.iterrows():
    B.add_edge(row[PRIMARY], row[SECONDARY], weight=1)

B.add_nodes_from(primary_col, bipartite=0)
B.add_nodes_from(secondary_col, bipartite=1)
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(15, 15))
pos = nx.drawing.layout.bipartite_layout(B, primary_col)
nx.draw_networkx(B, pos=pos, node_size=150, width=0.2, node_color='red')
```

Hình 6. Code hiển thị đồ thị 2 phía

2.2.2. Đồ thị 1 phía

- Node: danh mục các sản phẩm (Category Name).
- Edge: 2 loại sản phẩm cùng được tiến hành giao từ 1 khu vực sẽ được nối với nhau tạo thành cạnh, ý nghĩa nói lên với cùng 1 khu vực tiến hành giao nhiều loại sản phẩm.
- Weight: trọng số là số khu vực trùng nhau đã tiến hành giao 2 loại sản phẩm.



Hình 7. Đồ thị 1 phía

- Code hiển thị đồ thị 1 phía:

```
G = bipartite.weighted_projected_graph(B, primary_col)
node_labels = dict(zip(primary_col, primary_col))
```

```
plt.figure(figsize=(15, 7))
pos = nx.spring_layout(G)

nx.draw_networkx_nodes(G, pos, nodelist=primary_col, node_color='yellow')
nx.draw_networkx_edges(G, pos, edge_color='grey', alpha=0.2)
nx.draw_networkx_labels(G, pos, labels=node_labels)
plt.show()
```

Hình 8. Code hiển thị đồ thị 1 phía

IV. THUẬT TOÁN PHÁT HIỆN CỘNG ĐỒNG

1. Định nghĩa các hàm hiển thị kết quả phân cụm

- Hàm lấy đặc điểm chung của các node trong 1 cụm:

```
df_groupby_category = df.groupby([SECONDARY])[PRIMARY].apply(lambda x: sorted(set(x)))
df_groupby_category = df_groupby_category.reset_index()
# print(df_groupby_category.iloc[14][1])
df_groupby_category.head()
```

	Order Region	Category Name
0	Canada	[Accessories, Baseball & Softball, Boxing & MM...
1	Caribbean	[Accessories, As Seen on TV!, Baseball & Soft...
2	Central Africa	[Accessories, Baseball & Softball, Boxing & MM...
3	Central America	[Accessories, As Seen on TV!, Baseball & Soft...
4	Central Asia	[Accessories, Boxing & MMA, Camping & Hiking, ...

```
def get_cluster_common_values(cluster):
    commons = []
    for index, row in df_groupby_category.iterrows():
        if set(cluster).issubset(row[PRIMARY]):
            commons.append(row[SECONDARY])
    return commons
```

Hình 9. Hàm lấy đặc điểm chung của 1 cụm

- Hàm in ra các cộng đồng đã phát hiện được cùng điểm chung của các node trong chúng:

```
def print_communities(node_groups):
    print('Number of communities:', len(node_groups))
    for index, cluster in enumerate(node_groups):
        cluster = sorted(cluster)
        common_values = sorted(get_cluster_common_values(cluster))

        print(f'\nCluster {index}:')
        print(f"- {len(cluster)} Nodes: {', '.join(cluster)}")
        print(f"- {len(common_values)} Common values: {', '.join(common_values)}")
```

Hình 10. Hàm in ra các cộng đồng cùng điểm chung của các node trong chúng

- Hàm trực quan hóa các cộng đồng:

```
import matplotlib.cm as cm
def plot_communities(nodes, labels):
    cmap = cm.get_cmap('autumn', max(labels) + 1)
    pos = nx.spring_layout(G)

    plt.figure(figsize=(15, 7))
    nx.draw_networkx_nodes(G, pos, nodes, cmap=cmap, node_color=labels)
    nx.draw_networkx_edges(G, pos, edge_color='grey', alpha=0.2)
    nx.draw_networkx_labels(G, pos)
    plt.show()
```

Hình 11. Hàm trực quan hóa các cộng đồng

2. Thuật toán Louvain

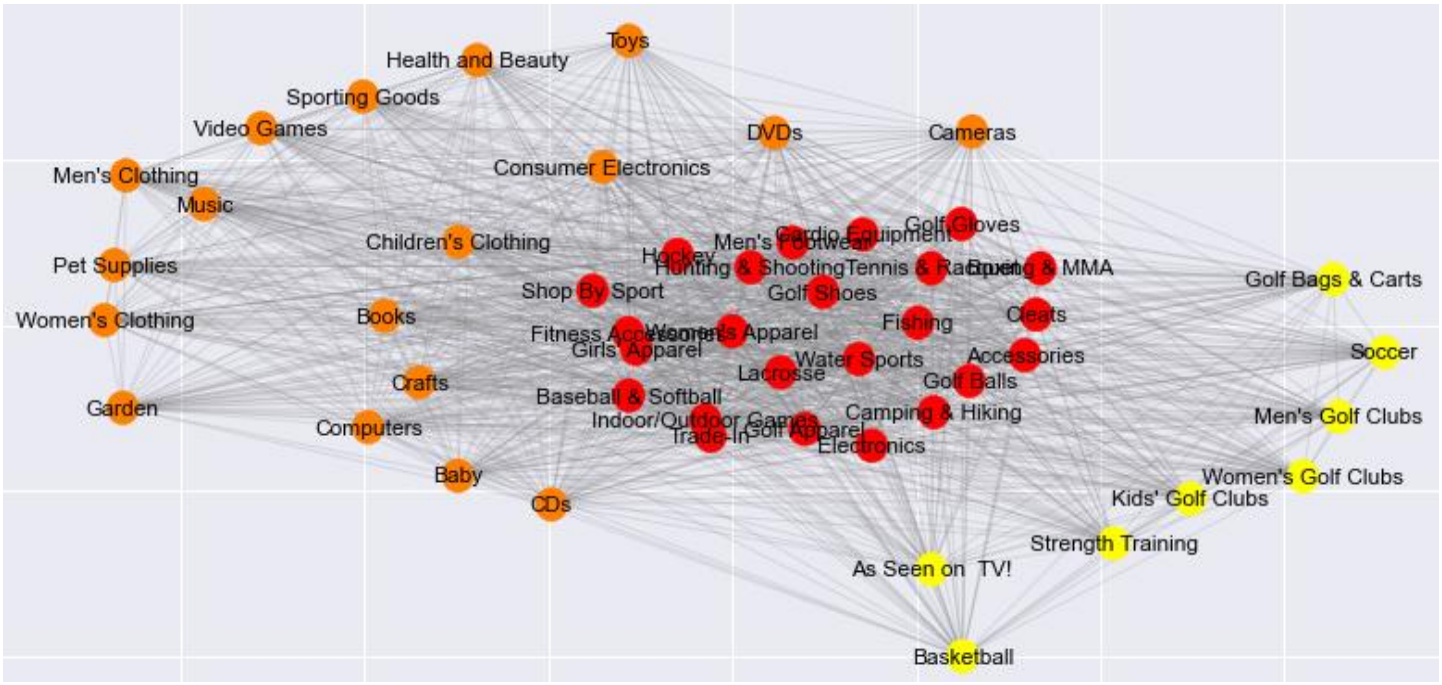
2.1. Code chạy thuật toán

```
import community.community_louvain as community_louvain
partition = community_louvain.best_partition(G)
louvain_node_groups = [[] for _ in set(partition.values())]

for node, cluster in sorted(partition.items()):
    louvain_node_groups[cluster].append(node)
```

```
print_communities(louvain_node_groups)
plot_communities(partition.keys(), list(partition.values()))
```

Hình 12. Code chạy thuật toán Louvain

2.2. Đồ thị phân cụm



Hình 13. Đồ thị phân cụm sử dụng Louvain

⇨ Dễ thấy thuật toán rút ra được 3 cụm

2.3. Ý nghĩa các cụm

❖ Cụm thứ 0:

- Gồm **24 Node**: Accessories, Baseball & Softball, Boxing & MMA, Camping & Hiking, Cardio Equipment, Cleats, Electronics, Fishing, Fitness Accessories, Girls' Apparel, Golf Apparel, Golf Balls, Golf Gloves, Golf Shoes, Hockey, Hunting & Shooting, Indoor/Outdoor Games, Lacrosse, Men's Footwear, Shop By Sport, Tennis & Racquet, Trade-In, Water Sports, Women's Apparel.
- Ý nghĩa: những loại sản phẩm được tiến hành giao ở cả **22 khu vực** gồm Canada, Caribbean, Central Africa, Central America, East Africa, East of USA, Eastern Asia, Eastern Europe, North Africa, Northern Europe, Oceania, South America, South Asia, South of USA, Southeast Asia, Southern Africa, Southern Europe, US Center, West Africa, West Asia, West of USA, Western Europe.

❖ Cụm thứ 1:

- Gồm **18 Node**: Baby, Books, CDs, Cameras, Children's Clothing, Computers, Consumer Electronics, Crafts, DVDs, Garden, Health and Beauty, Men's Clothing, Music, Pet Supplies, Sporting Goods, Toys, Video Games, Women's Clothing.
- Ý nghĩa: những loại sản phẩm được tiến hành giao ở cả **3 khu vực** gồm Oceania, South Asia, Southeast Asia.

❖ Cụm thứ 2:

- Gồm **8 Node**: As Seen on TV!, Basketball, Golf Bags & Carts, Kids' Golf Clubs, Men's Golf Clubs, Soccer, Strength Training, Women's Golf Clubs.
- Ý nghĩa: những loại sản phẩm được tiến hành giao ở cả **3 khu vực** gồm Northern Europe, Southern Europe, Western Europe.

2.4. Kết luận

- Nếu nhìn bằng mắt thường ta cũng dễ dàng nhận ra đồ thị có thể có 3 cụm. Louvain đã phát hiện được 3 cụm đó khá rõ rệt với nhau. Như vậy chứng tỏ thuật toán Louvain đã phân cụm khá tốt.

3. Thuật toán K-Means

3.1. Code chạy thuật toán

3.1.1. Chuyển đổi đồ thị thành ma trận kề

```
from sklearn.cluster import KMeans
from scipy.spatial.distance import cdist
adj_matrix = nx.to_numpy_array(G)
```

Hình 14. Chuyển đổi đồ thị thành ma trận kề làm đầu vào cho K-Means

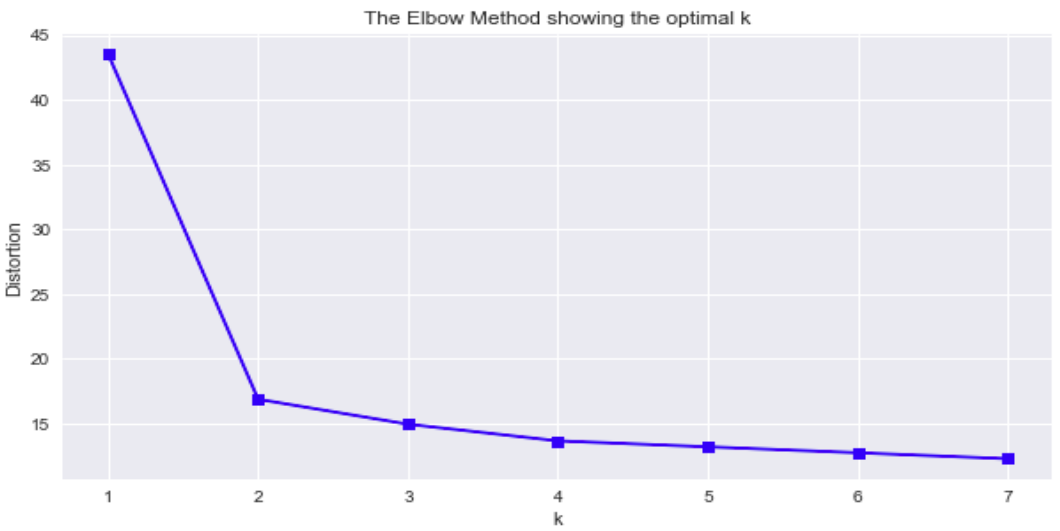
3.1.2. Chọn ra số cụm bằng Elbow Method

```
import numpy as np
distortions = []
K = range(1, 8)

for k in K:
    kmean_model = KMeans(n_clusters=k)
    kmean_model.fit(adj_matrix)
    dist = sum(np.min(cdist(adj_matrix, kmean_model.cluster_centers_, 'euclidean'), axis=1))
    distortions.append(dist / adj_matrix.shape[0])
```

Hình 15. Code triển khai Elbow Method cho thuật toán K-Means

- Tính khoảng cách từng phần tử trong ma trận kề (50 x 50) với các tâm cụm (k x 50).
- Ma trận (50 x k) tương ứng với 50 loại sản phẩm, với hàng là các khoảng cách giữa mỗi phần tử trong ma trận kề với các tâm cụm.
- Lấy tổng các khoảng cách của mỗi phần tử trong ma trận kề với tâm cụm mà tại đó có khoảng cách giữa 2 bên là nhỏ nhất / số lượng phần tử trong ma trận kề.
- Cuối cùng, tính tổng biến thiên khoảng cách nhỏ nhất của trong cụm.

```
plt.figure(figsize=(10, 5))
plt.plot(K, distortions, 'bs-')
plt.xlabel('k')
plt.ylabel('Distortion')
plt.title('The Elbow Method showing the optimal k')
plt.show()
```



Hình 16. Chọn ra số cụm k cho thuật toán K-Means bằng Elbow Method

3.1.3. Huấn luyện với số cụm đã chọn

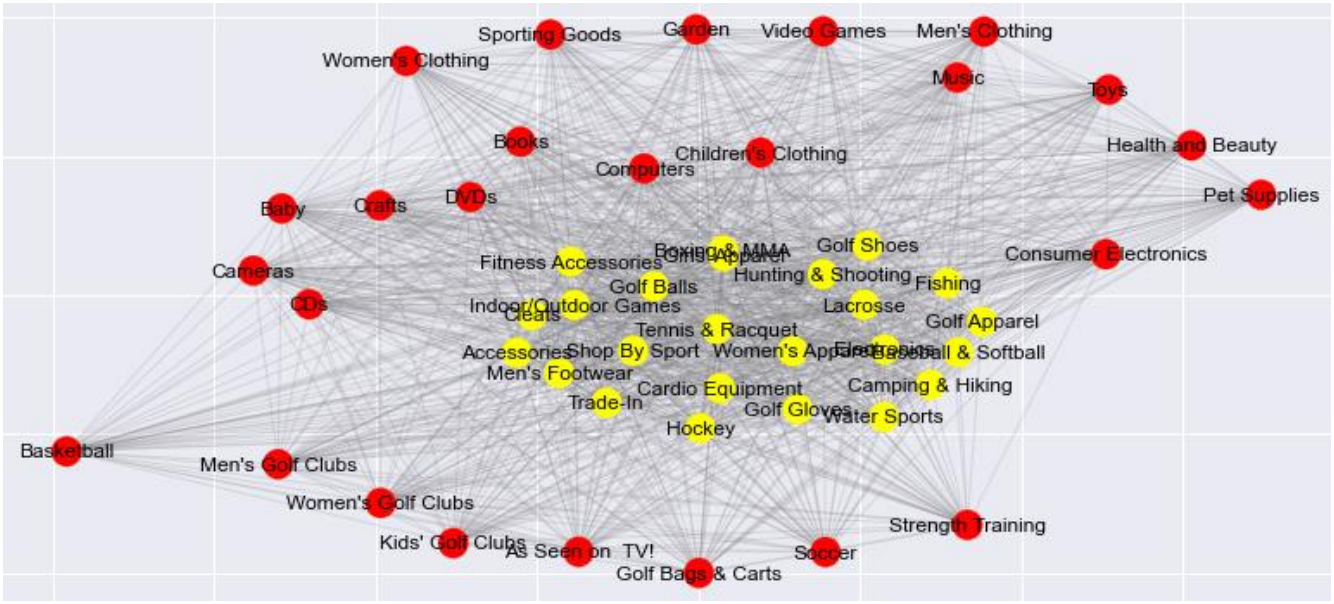
- Chọn số cụm k là 2 để huấn luyện cũng như sẽ là số lượng centroid sẽ tạo ra.
- Tiến hành huấn luyện và hiển thị kết quả phân cụm với đầu vào là ma trận kề đã có.

```
kmeans = KMeans(n_clusters=2)
kmeans.fit(adj_matrix)

kmeans_node_groups = [[] for _ in range(kmeans.n_clusters)]
for node, cluster in zip(G.nodes(), kmeans.labels_):
    kmeans_node_groups[cluster].append(node)
```

```
print_communities(kmeans_node_groups)
plot_communities(G.nodes(), kmeans.labels_)
```

Hình 17. Chạy K-Means với đầu vào là ma trận kề cùng số cụm k đã chọn

3.2. Đồ thị phân cụm



Hình 18. Đồ thị phân cụm sử dụng K-Means

3.3. Ý nghĩa các cụm

❖ Cụm thứ 0:

- Gồm **26 Node**: As Seen on TV!, Baby, Basketball, Books, CDs, Cameras, Children's Clothing, Computers, Consumer Electronics, Crafts, DVDs, Garden, Golf Bags & Carts, Health and Beauty, Kids' Golf Clubs, Men's Clothing, Men's Golf Clubs, Music, Pet Supplies, Soccer, Sporting Goods, Strength Training, Toys, Video Games, Women's Clothing, Women's Golf Clubs.
- Ý nghĩa: các node trong cụm này **không** có điểm chung gì đồng thời với tất cả các node khác. Đây có thể đơn giản chỉ là phần còn lại sau khi phát hiện được cụm 1.

❖ Cụm thứ 1:

- Gồm **24 Node**: Accessories, Baseball & Softball, Boxing & MMA, Camping & Hiking, Cardio Equipment, Cleats, Electronics, Fishing, Fitness Accessories, Girls' Apparel, Golf Apparel, Golf Balls, Golf Gloves, Golf Shoes, Hockey, Hunting & Shooting, Indoor/Outdoor Games, Lacrosse, Men's Footwear, Shop By Sport, Tennis & Racquet, Trade-In, Water Sports, Women's Apparel.
- Ý nghĩa: những loại sản phẩm được tiến hành giao ở cả **22 khu vực** gồm Canada, Caribbean, Central Africa, Central America, East Africa, East of USA, Eastern Asia, Eastern Europe, North Africa, Northern Europe, Oceania, South America, South Asia, South of USA, Southeast Asia, Southern Africa, Southern Europe, US Center, West Africa, West Asia, West of USA, Western Europe.

3.4. Kết luận

- Từ kết quả có thể thấy K-Means chỉ phân biệt được các node có phân bố dạng chụm lại với nhau tạo thành 1 cụm (màu vàng) nhưng không phân biệt được hay tìm được các đặc điểm chung của các node có phân bố dạng đường vòng (màu đỏ).
- Nguyên nhân có thể do khi sử dụng thuật toán K-Means thì mỗi điểm dữ liệu được gán 1 cách dứt khoát thuộc 1 trung tâm cụm. Ngoài ra, K-Means không tính đến covariance (hiệp phương sai) để thể hiện tính liên quan của dữ liệu. Nếu ta có 2 điểm cách đều nhau từ trung tâm cụm nhưng 1 điểm theo xu hướng này và điểm kia thì không, K-Means sẽ coi chúng là như nhau, vì nó sử dụng khoảng cách Euclide.

4. Gaussian Mixture Model

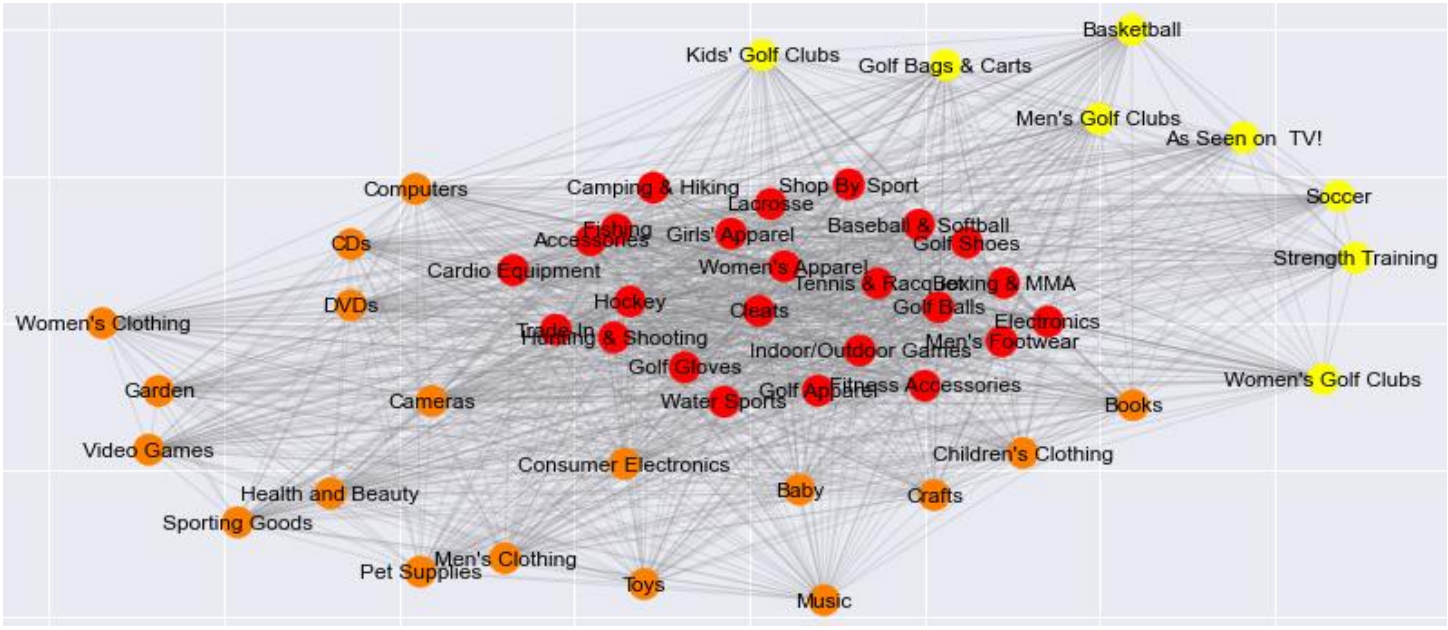
4.1. Code chạy thuật toán

```
from sklearn.mixture import GaussianMixture
n_clusters = 3
gmm = GaussianMixture(n_components=n_clusters)
gmm.fit(adj_matrix)

labels = gmm.predict(adj_matrix)
gmm_node_groups = [[] for _ in range(n_clusters)]
for node, cluster in zip(G.nodes(), labels):
    gmm_node_groups[cluster].append(node)
```

```
print_communities(gmm_node_groups)
plot_communities(G.nodes(), labels)
```

Hình 19. Code chạy mô hình Gaussian Mixture

4.2. Đồ thị phân cụm



Hình 20. Đồ thị phân cụm sử dụng GMM

4.3. Ý nghĩa các cụm

❖ Cụm thứ 0:

- Gồm **24 Node**: Accessories, Baseball & Softball, Boxing & MMA, Camping & Hiking, Cardio Equipment, Cleats, Electronics, Fishing, Fitness Accessories, Girls' Apparel, Golf Apparel, Golf Balls, Golf Gloves, Golf Shoes, Hockey, Hunting & Shooting, Indoor/Outdoor Games, Lacrosse, Men's Footwear, Shop By Sport, Tennis & Racquet, Trade-In, Water Sports, Women's Apparel.
- Ý nghĩa: những loại sản phẩm được tiến hành giao ở cả **22 khu vực** gồm Canada, Caribbean, Central Africa, Central America, East Africa, East of USA, Eastern Asia, Eastern Europe, North Africa, Northern Europe, Oceania, South America, South Asia, South of USA, Southeast Asia, Southern Africa, Southern Europe, US Center, West Africa, West Asia, West of USA, Western Europe.

❖ Cụm thứ 1:

- Gồm **18 Node**: Cleats, Baby, Books, CDs, Cameras, Children's Clothing, Computers, Consumer Electronics, Crafts, DVDs, Garden, Health and Beauty, Men's Clothing, Music, Pet Supplies, Sporting Goods, Toys, Video Games, Women's Clothing.

- Ý nghĩa: những loại sản phẩm được tiến hành giao ở cả **3 khu vực** gồm Oceania, South Asia, Southeast Asia.

❖ Cụm thứ 2:

- Gồm **8 Node**: As Seen on TV!, Basketball, Golf Bags & Carts, Kids' Golf Clubs, Men's Golf Clubs, Soccer, Strength Training, Women's Golf Clubs.
- Ý nghĩa: những loại sản phẩm được tiến hành giao ở cả **3 khu vực** gồm Northern Europe, Southern Europe, Western Europe.

4.4. Kết luận

- GMM là một mô hình xác suất giả định tất cả các điểm dữ liệu được tạo ra từ một hỗn hợp của 1 số hữu hạn của các phân phối Gaussian với các tham số không xác định và là phần mở rộng các ý tưởng đằng sau K-Means, vì vậy GMM đã khắc phục được các nhược điểm của K-Means và đã cho ra kết quả phân cụm tương tự thuật toán Louvain.

V. XẾP HẠNG CÁC LOẠI SẢN PHẨM

1. Định nghĩa các hàm hiển thị kết quả độ đo

- Hàm in kết quả độ đo đã sắp xếp giảm dần:

```
def print_centrality(centrality, name):
    result = pd.DataFrame(centrality.items(), columns=['Category Name', name])
    result.sort_values(name, ascending=False, inplace=True)
    print(result.to_records(index=False).tolist())
    return result
```

Hình 21. Hàm in kết quả độ đo

- Hàm trực quan hóa kết quả độ đo:

```
def plot_centrality(centrality):
    plt.figure(figsize=(15, 7))
    node_color = [centrality[i] for i in centrality.keys()]

    cmap = plt.cm.ScalarMappable(
        cmap = 'Wistia',
        norm = plt.Normalize(vmin=min(node_color), vmax=max(node_color))
    )
    cmap.set_array([])
    plt.colorbar(cmap)

    pos = nx.spring_layout(G)
    nx.draw_networkx_nodes(G, pos, node_color=node_color, cmap='Wistia')
    nx.draw_networkx_edges(G, pos, edge_color='grey', alpha=0.2)
    nx.draw_networkx_labels(G, pos)
    plt.show()
```

Hình 22. Hàm trực quan hóa kết quả độ đo

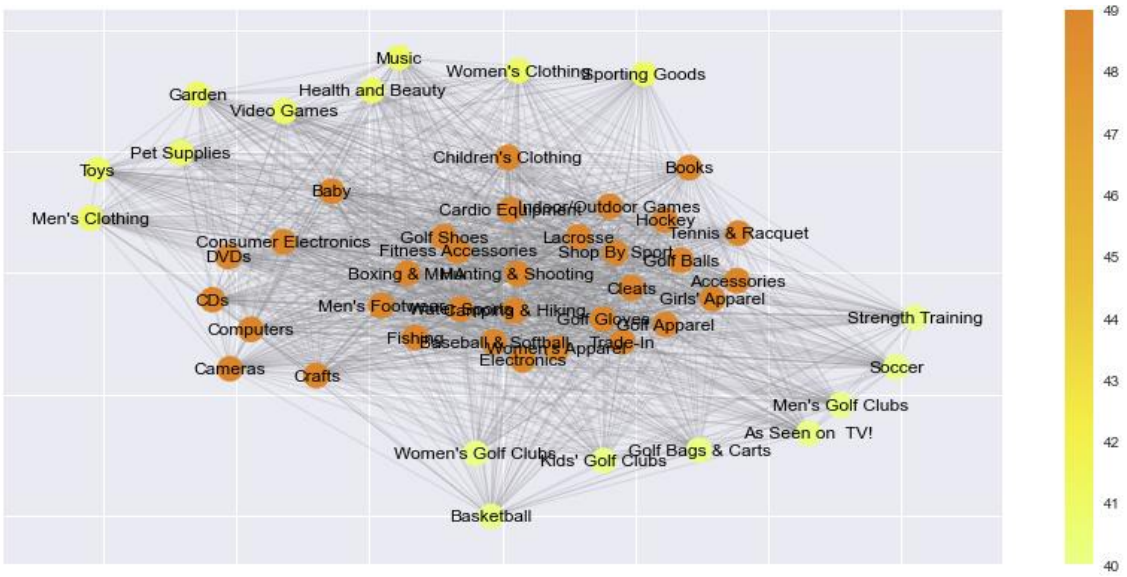
2. Degree Centrality

2.1. Code chạy độ đo

```
degree = dict(nx.degree(G))
plot_centrality(degree)
print_centrality(degree, 'Degree Centrality').head()
```

Hình 23. Code chạy Degree Centrality

2.2. Kết quả độ đo




```
[('Indoor/Outdoor Games', 49), ('Golf Balls', 49), ('Golf Shoes', 49), ('Crafts', 49), ('Golf Gloves', 49), ('Tennis & Racque
t', 49), ('Fitness Accessories', 49), ('Cleats', 49), ("Children's Clothing", 49), ('Golf Apparel', 49), ('Lacrosse', 49), ('Ba
by', 49), ('Fishing', 49), ('Books', 49), ('DVDs', 49), ('CDs', 49), ('Accessories', 49), ("Girls' Apparel", 49), ('Hockey', 4
9), ('Shop By Sport', 49), ('Computers', 49), ('Cameras', 49), ('Consumer Electronics', 49), ('Camping & Hiking', 49), ("Men's
Footwear", 49), ('Baseball & Softball', 49), ('Hunting & Shooting', 49), ('Water Sports', 49), ('Trade-In', 49), ('Cardio Equip
ment', 49), ('Boxing & MMA', 49), ('Electronics', 49), ("Women's Apparel", 49), ('Music', 41), ('Health and Beauty', 41), ('Toy
s', 41), ('Video Games', 41), ('Sporting Goods', 41), ('Pet Supplies', 41), ('Garden', 41), ("Men's Clothing", 41), ("Women's C
lothing", 41), ('Strength Training', 40), ('Golf Bags & Carts', 40), ('Soccer', 40), ("Women's Golf Clubs", 40), ("Men's Golf C
lubs", 40), ('Basketball', 40), ("Kids' Golf Clubs", 40), ('As Seen on  TV!', 40)]
```

Hình 24. Kết quả Degree Centrality

- Nhận xét: Degree của 1 node thể hiện số lượng node mà 1 node nhất định kết nối. Loại sản phẩm có Degree càng cao thì càng có nhiều liên kết, nghĩa là các khu vực tiến hành giao 1 loại sản phẩm cũng sẽ tiến hành giao các loại sản phẩm khác. Như trong hình 24, ta có thể thấy các node ở ngoài rìa sẽ có Degree thấp hơn.

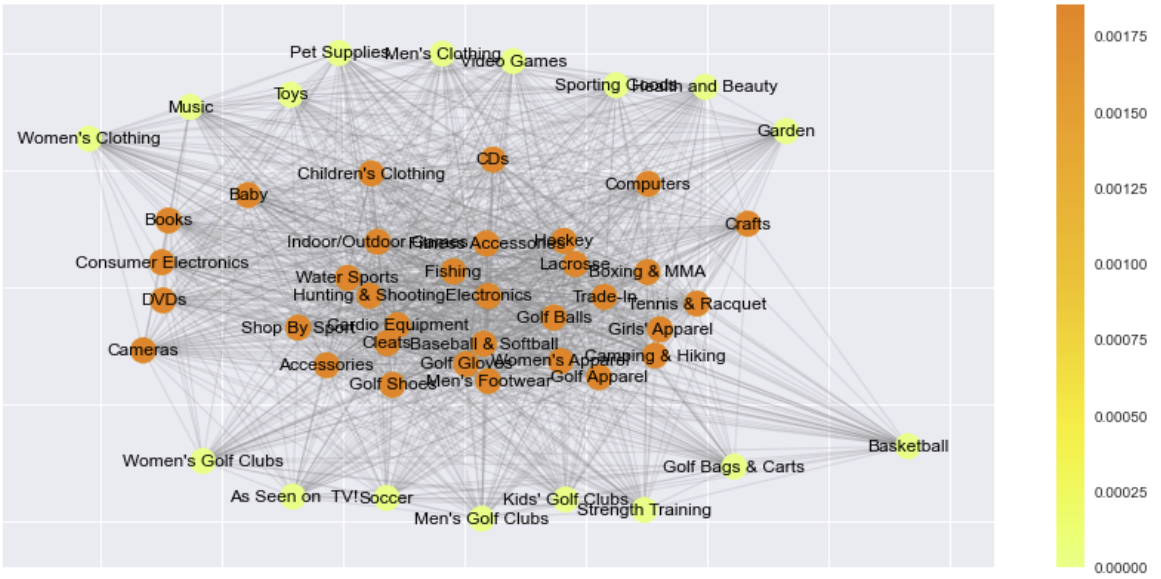
3. Betweenness Centrality

3.1. Code chạy độ đo

```
betweenness = nx.betweenness_centrality(G)
plot_centrality(betweenness)
print_centrality(betweenness, 'Betweenness Centrality').head()
```

Hình 25. Code chạy Betweenness Centrality

3.2. Kết quả độ đo




```
[('Indoor/Outdoor Games', 0.0018552875695732828), ("Girls' Apparel", 0.0018552875695732828), ('Golf Apparel', 0.0018552875695732828), ('Tennis & Racquet', 0.0018552875695732828), ('Fitness Accessories', 0.0018552875695732828), ('Cleats', 0.0018552875695732828), ('Golf Balls', 0.0018552875695732828), ('Golf Shoes', 0.0018552875695732828), ("Children's Clothing", 0.0018552875695732828), ('Lacrosse', 0.0018552875695732828), ('Baby', 0.0018552875695732828), ('Fishing', 0.0018552875695732828), ('Books', 0.0018552875695732828), ('DVDs', 0.0018552875695732828), ('CDs', 0.0018552875695732828), ('Hockey', 0.0018552875695732828), ('Golf Gloves', 0.0018552875695732828), ('Crafts', 0.0018552875695732828), ('Accessories', 0.0018552875695732828), ('Hunting & Shooting', 0.0018552875695732828), ('Electronics', 0.0018552875695732828), ('Shop By Sport', 0.0018552875695732828), ("Women's Apparel", 0.0018552875695732828), ('Computers', 0.0018552875695732828), ('Cameras', 0.0018552875695732828), ('Consumer Electronics', 0.0018552875695732828), ('Camping & Hiking', 0.0018552875695732828), ("Men's Footwear", 0.0018552875695732828), ('Baseball & Softball', 0.0018552875695732828), ('Water Sports', 0.0018552875695732828), ('Trade-In', 0.0018552875695732828), ('Cardio Equipment', 0.0018552875695732828), ('Boxing & MMA', 0.0018552875695732828), ('Golf Bags & Carts', 0.0), ('Music', 0.0), ("Women's Golf Clubs", 0.0), ('Toys', 0.0), ("Men's Golf Clubs", 0.0), ('Video Games', 0.0), ('Sporting Goods', 0.0), ('Health and Beauty', 0.0), ('Pet Supplies', 0.0), ('Garden', 0.0), ('Strength Training', 0.0), ("Men's Clothing", 0.0), ("Women's Clothing", 0.0), ('Soccer', 0.0), ('Basketball', 0.0), ("Kids' Golf Clubs", 0.0), ('As Seen on  TV!', 0.0)]
```

Hình 26. Kết quả Betweenness Centrality

- Nhận xét: Betweenness Centrality giữa các nút xác định mức độ "ở giữa" với các nút khác. Phép đo này tính toán các đường đi ngắn nhất giữa tất cả các node và chỉ định cho mỗi node 1 phép đo dựa trên số lượng đường đi ngắn nhất đi qua node đích. Loại sản phẩm có Betweenness Centrality cao hơn sẽ có nhiều quyền kiểm soát hơn đối với mạng vì nhiều thông tin hơn sẽ đi qua node đó. Như trong hình 26, ta có thể thấy các node màu cam có Betweenness Centrality cao nhất, nghĩa là nó quan trọng trong việc kết nối giữa các node, các node muốn kết nối với nhau phải thông qua nó.

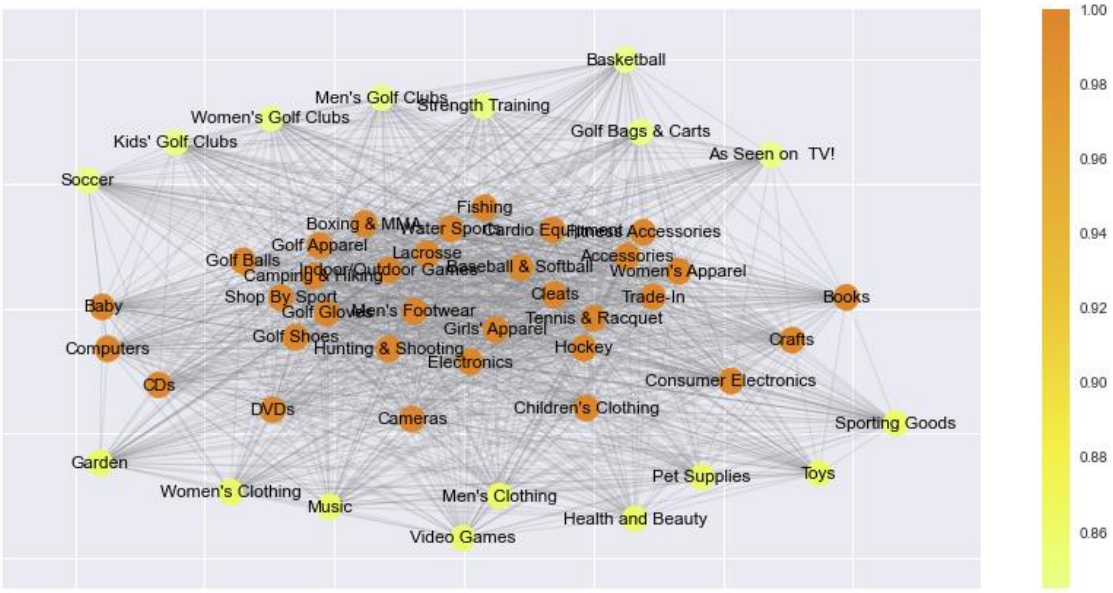
4. Closeness Centrality

4.1. Code chạy độ đo

```
closeness = nx.closeness_centrality(G)
plot_centrality(closeness)
print_centrality(closeness, 'Closeness Centrality').head()
```

Hình 27. Code chạy Closeness Centrality

4.2. Kết quả độ đo




```
[('Indoor/Outdoor Games', 1.0), ('Golf Balls', 1.0), ('Golf Shoes', 1.0), ('Crafts', 1.0), ('Golf Gloves', 1.0), ('Tennis & Racquet', 1.0), ('Fitness Accessories', 1.0), ('Cleats', 1.0), ("Children's Clothing", 1.0), ('Golf Apparel', 1.0), ('Lacrosse', 1.0), ('Baby', 1.0), ('Fishing', 1.0), ('Books', 1.0), ('DVDs', 1.0), ('CDs', 1.0), ('Accessories', 1.0), ("Girls' Apparel", 1.0), ('Hockey', 1.0), ('Shop By Sport', 1.0), ('Computers', 1.0), ('Cameras', 1.0), ('Consumer Electronics', 1.0), ('Camping & Hiking', 1.0), ("Men's Footwear", 1.0), ('Baseball & Softball', 1.0), ('Hunting & Shooting', 1.0), ('Water Sports', 1.0), ('Trade-In', 1.0), ('Cardio Equipment', 1.0), ('Boxing & MMA', 1.0), ('Electronics', 1.0), ("Women's Apparel", 1.0), ('Music', 0.8596491228070176), ('Health and Beauty', 0.8596491228070176), ('Toys', 0.8596491228070176), ('Video Games', 0.8596491228070176), ('Sporting Goods', 0.8596491228070176), ('Pet Supplies', 0.8596491228070176), ('Garden', 0.8596491228070176), ("Men's Clothing", 0.8596491228070176), ("Women's Clothing", 0.8596491228070176), ('Strength Training', 0.8448275862068966), ('Golf Bags & Carts', 0.8448275862068966), ('Soccer', 0.8448275862068966), ("Women's Golf Clubs", 0.8448275862068966), ("Men's Golf Clubs", 0.8448275862068966), ('Basketball', 0.8448275862068966), ("Kids' Golf Clubs", 0.8448275862068966), ('As Seen on  TV!', 0.8448275862068966)]
```


Hình 28. Kết quả Closeness Centrality

- Nhận xét: Closeness Centrality là 1 cách phát hiện các nút có thể phân phối luồng đi 1 cách hiệu quả qua mạng, node sẽ quan trọng là node ở gần với các node khác. Kết quả của Closeness Centrality thể hiển top độ gần các node đến tất cả các node trong mạng. Nó cũng đồng nghĩa với việc top các loại sản phẩm được tiến hành giao nhiều nhất.

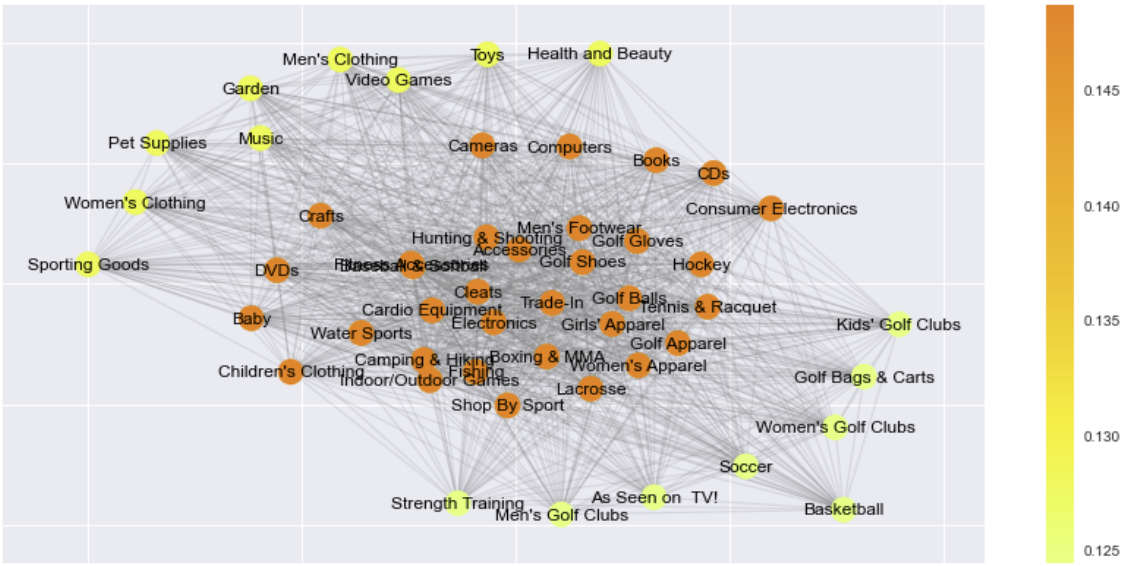
5. Eigenvector Centrality

5.1. Code chạy độ đo

```
eigenvector = nx.eigenvector_centrality(G)
plot_centrality(eigenvector)
print_centrality(eigenvector, 'Eigenvector Centrality').head()
```

Hình 29. Code chạy Eigenvector Centrality

5.2. Kết quả độ đo




```
[('Indoor/Outdoor Games', 0.14867064721396608), ("Children's Clothing", 0.14867064721396608), ('Crafts', 0.14867064721396608),
('Golf Gloves', 0.14867064721396608), ('Tennis & Racquet', 0.14867064721396608), ('Fitness Accessories', 0.14867064721396608),
('Cleats', 0.14867064721396608), ('Golf Balls', 0.14867064721396608), ('Lacrosse', 0.14867064721396608), ("Girls' Apparel", 0.1
4867064721396608), ('Baby', 0.14867064721396608), ('Fishing', 0.14867064721396608), ('Books', 0.14867064721396608), ('DVDs', 0.
14867064721396608), ('CDs', 0.14867064721396608), ('Hockey', 0.14867064721396608), ('Golf Shoes', 0.14867064721396608), ('Acces
sories', 0.14867064721396608), ('Golf Apparel', 0.14867064721396608), ('Water Sports', 0.14867064721396608), ('Shop By Sport',
0.14867064721396608), ("Women's Apparel", 0.14867064721396608), ('Electronics', 0.14867064721396608), ('Boxing & MMA', 0.148670
64721396608), ('Cardio Equipment', 0.14867064721396608), ('Trade-In', 0.14867064721396608), ('Hunting & Shooting', 0.1486706472
1396608), ('Baseball & Softball', 0.14867064721396608), ("Men's Footwear", 0.14867064721396608), ('Camping & Hiking', 0.1486706
4721396608), ('Consumer Electronics', 0.14867064721396608), ('Cameras', 0.14867064721396608), ('Computers', 0.1486706472139660
8), ('Video Games', 0.12767929833125463), ('Music', 0.12767929833125463), ('Garden', 0.12767929833125463), ('Toys', 0.127679298
33125463), ('Pet Supplies', 0.12767929833125463), ('Health and Beauty', 0.12767929833125463), ('Sporting Goods', 0.127679298331
2546), ("Men's Clothing", 0.1276792983312546), ("Women's Clothing", 0.1276792983312546), ('Strength Training', 0.12444087284685
672), ('Golf Bags & Carts', 0.12444087284685672), ('Soccer', 0.12444087284685672), ("Women's Golf Clubs", 0.12444087284685672),
("Men's Golf Clubs", 0.12444087284685672), ('Basketball', 0.12444087284685672), ("Kids' Golf Clubs", 0.12444087284685672), ('As
Seen on  TV!', 0.12444087284685672)]
```

Hình 30. Kết quả Eigenvector Centrality

- Nhận xét: Eigenvector Centrality đo lường tầm quan trọng của 1 node trong mạng đồng thời xem xét tầm quan trọng của các node lân cận. Các kết nối đến các node trung tâm của eigenvector có điểm số cao đóng góp nhiều hơn vào điểm số chung so với các kết nối ngang bằng với các nút có điểm số thấp. Nói cách khác, 1 node có nhiều kết nối có thể có điểm Eigenvector thấp nếu tất cả các kết nối của nó đều có các node có điểm thấp. Loại sản phẩm có điểm Eigenvector cao khi nó được kết nối với nhiều node có điểm cao. Trong hình 30, ta có thể thấy các node màu cam có ảnh hưởng nhất mạng.

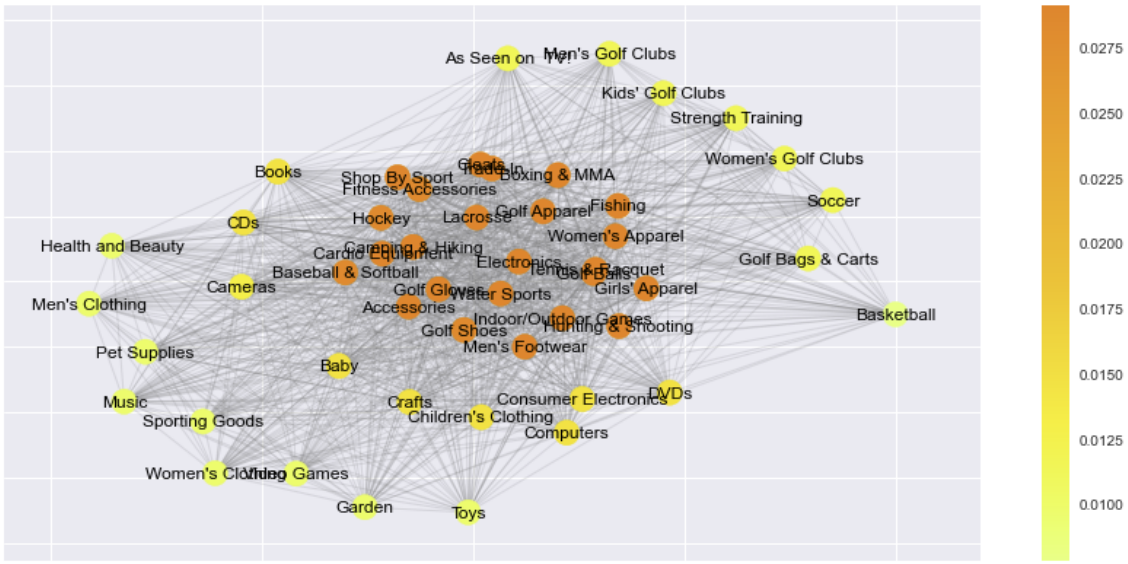
6. PageRank

6.1. Code chạy độ đo

```
1 pagerank = nx.pagerank(G)
2 plot_centrality(pagerank)
3 print_centrality(pagerank, 'PageRank').head()
```

Hình 31. Code chạy PageRank

6.2. Kết quả độ đo




```
[('Indoor/Outdoor Games', 0.02910004330793566), ("Men's Footwear", 0.02910004330793566), ('Fishing', 0.02910004330793566), ('La
crosse', 0.02910004330793566), ('Golf Balls', 0.02910004330793566), ('Cleats', 0.02910004330793566), ('Tennis & Racquet', 0.029
10004330793566), ('Golf Apparel', 0.02910004330793566), ('Accessories', 0.02910004330793566), ('Camping & Hiking', 0.0291000433
0793566), ("Girls' Apparel", 0.02910004330793566), ('Cardio Equipment', 0.02910004330793566), ('Water Sports', 0.02910004330793
566), ('Shop By Sport', 0.02910004330793566), ("Women's Apparel", 0.02910004330793566), ('Electronics', 0.02910004330793566),
('Boxing & MMA', 0.02910004330793566), ('Hunting & Shooting', 0.02910004330793566), ('Trade-In', 0.02910004330793566), ('Golf G
loves', 0.02910004330793566), ('Hockey', 0.02839302811074137 5), ('Golf Shoes', 0.02839302811074137 5), ('Fitness Accessories',
0.02839302811074137 5), ('Baseball & Softball', 0.02839302811074137 5), ('Crafts', 0.01460032971973412), ('Computers', 0.01460032
971973412), ("Children's Clothing", 0.01460032971973412), ('Baby', 0.01460032971973412), ('Consumer Electronics', 0.01460032971
973412), ('Books', 0.01460032971973412), ('DVDs', 0.01460032971973412), ('CDs', 0.01460032971973412), ('Cameras', 0.01293948539
0217484), ("Women's Golf Clubs", 0.01134315515704318), ("Men's Golf Clubs", 0.01134315515704318), ('Golf Bags & Carts', 0.01134
315515704318), ('As Seen on  TV!', 0.01134315515704318), ('Strength Training', 0.01134315515704318), ('Soccer', 0.0113431551570
4318), ("Kids' Golf Clubs", 0.01134315515704318), ('Pet Supplies', 0.009718050588254678), ('Health and Beauty', 0.0097180505882
54678), ('Music', 0.009718050588254678), ('Video Games', 0.009718050588254678), ('Garden', 0.009718050588254678), ("Men's Cloth
ing", 0.009718050588254678), ("Women's Clothing", 0.009718050588254678), ('Toys', 0.009718050588254678), ('Sporting Goods', 0.0
09718050588254678), ('Basketball', 0.007820356856636094)]
```

Hình 32. Kết quả PageRank

- Nhận xét: 1 biến thể phổ biến của Eigenvector Centrality là PageRank của Google. Về bản chất, PageRank là 1 biến thể của Eigenvector Centrality được chuẩn hóa kết hợp với các bước nhảy ngẫu nhiên. Do đó, không có gì ngạc nhiên khi Pagerank mang lại kết quả tương đương hoặc có thể tốt hơn cả Eigenvector. Như trong hình 31, ta có thể thấy PageRank đã có sự phân hóa rõ rệt hơn các node quan trọng trong mạng so với các độ đo trước.

VI. TÀI LIỆU THAM KHẢO

1. https://youtube.com/playlist?list=PLoROMvodv4rPLKxIpqhjhPgdQy7imNkDn
2. https://www.coursera.org/learn/python-social-network-analysis
3. https://github.com/Geometrein/helsinki-city-bikes