XU HƯỚNG HỌC MÔN HỌC TỰ CHỌN CỦA SINH VIÊN NGÀNH IT

CHƯƠNG 1. GIỚI THIỆU

1.1. Lý do chọn đề tài

Tại trường Đại học Công nghệ Thông tin (UIT), mỗi năm có khoảng 1800 sinh viên nhập học. Để hoàn thành chương trình đào tạo, ngoài các môn chuyên ngành bắt buộc, mỗi sinh viên được khuyến khích học 12 tín chỉ về môn học tự chọn của các ngành khác để có thể có được các kiến thức và kỹ năng của toàn bộ nhóm ngành công nghệ thông tin. Tuy nhiên, việc mở toàn bộ các môn học của tất cả các ngành là việc không cần thiết vì vấn đề chi phí và độ hiệu quả. Ngoài ra, do số lượng sinh viên ngày càng đông và đa dạng hơn (về trình độ, chuyên môn và mục tiêu) việc xác định xu hướng học tập các môn tự chọn là cần thiết để nhà trường có thể điều chỉnh số lượng lớp học cần mở. Trong đồ án này, tôi đã phân tích xu hướng học các môn tự chọn của sinh viên sử dụng các thuật toán gom cụm (Louvain, K – mean,..) và các độ đo (PageRank, Clossness Centrality,…) trên bộ dữ liệu về lịch sử đăng kí môn học của sinh viên UIT.

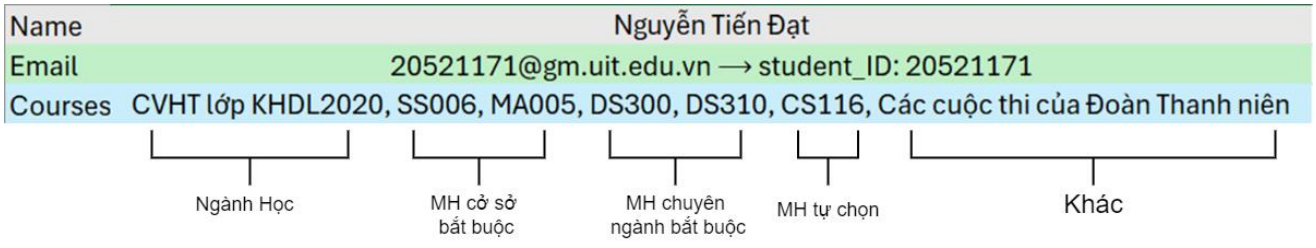
1.2. Xác định bài toán

Input: Bộ dữ liệu về lịch sử đăng kí môn học của sinh viên đã qua tiền xử lý.
Output: Các cộng đồng (cluster) và độ đo trên mạng xã hội biểu diễn các sinh viên

CHƯƠNG 2. DỮ LIỆU

2.1 Mô tả dữ liệu

Tập dữ liệu chúng tôi sử dụng để xây dựng đồ thị biểu diễn sinh viên được thu thập từ courses.uit.edu.vn. Tập dữ liệu này chứa lịch sử đăng ký môn học của 8800 sinh viên (bao gồm sinh viên đã tốt nghiệp và sinh viên đang trong chương trình học) tại Đại học Công nghệ Thông tin vào bất kỳ thời điểm nào bắt đầu từ năm 2016 đến năm 2023 (tương ứng với sinh viên từ khóa 11 đến khóa 16). Dữ liệu này bao gồm tên, email mà trường cung cấp cho sinh viên, và danh sách các lớp mà sinh viên đó đã đăng kí. Hình 1 là danh sách môn học mà một sinh viên đã đăng kí. Danh sách bao gồm 5 thành phần: lớp chuyên ngành, môn học cơ sở bắt buộc, môn học chuyên ngành bắt buộc, môn học tự chọn và khác.

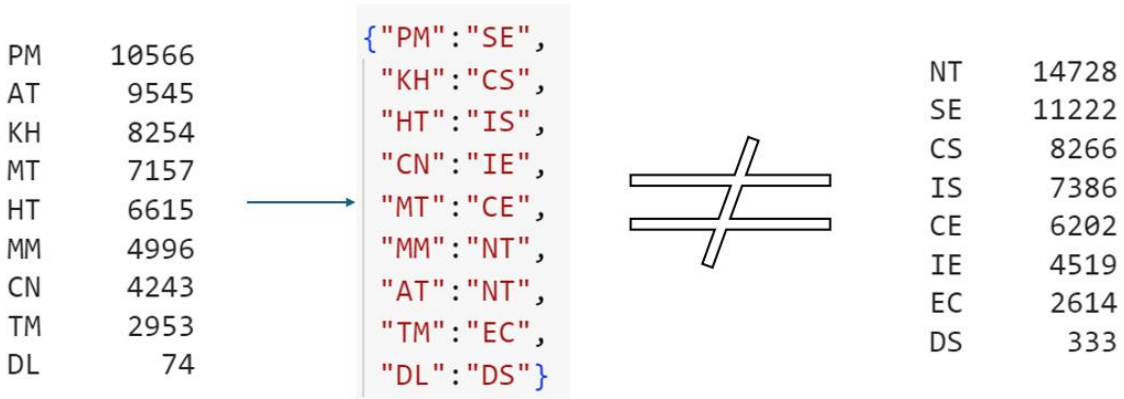


Hình 1. Các môn học mà 1 sinh viên đăng kí

2.2 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu chủ yếu liên quan đến việc tách và trích xuất thông tin cần thiết từ danh sách các môn sinh viên viên đã đăng kí. Do ảnh hưởng của việc cào dữ liệu từ website, bộ dữ liệu ban đầu có một số dòng chứa dữ liệu rỗng và dữ liệu không phải của sinh viên thì những dòng này sẽ được loại bỏ. Đối với lịch sử học tập mỗi sinh viên, tất cả các môn học cở sở bắt buộc (toàn bộ sinh viên phải học) sẽ được loại bỏ vì những môn học này *không có ý nghĩa trong việc phân biệt các sinh viên với nhau*.
Tiếp theo, để xác định được những môn chuyên ngành tự chọn và môn chuyên ngành bắt buộc, tôi sử dụng mã ngành mà sinh viên đó học để so sánh với mã của môn học. Nếu mã ngành của sinh viên khác với mã của môn học thì môn học đó được xem là môn tự chọn. Đối với những sinh viên chuyển ngành thì sinh viên đó sẽ có nhiều mã ngành được ghi lại trong lịch sử học tập vì vậy mã ngành cuối cùng mà sinh viên đó học sẽ được giữ lại để xác định ngành học cho sinh viên đó. Dữ liệu cuối cũng sẽ chỉ bao gồm những môn học tự chọn mà sinh viên đã đăng kí. Các thông tin cá nhân khác về sinh viên (tên, email,. . .) sẽ được bỏ qua.

Hình 2. Trích xuất ngành học của sinh viên

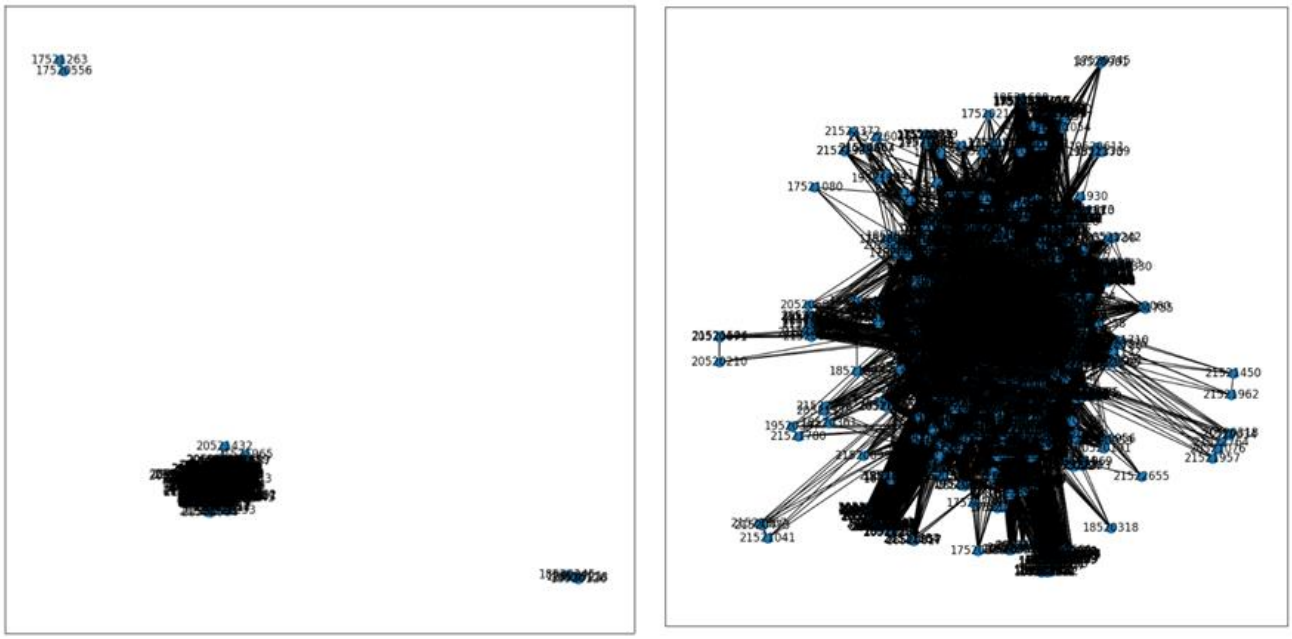


Hình 3. Filter những lượt đăng kí là môn học tự chọn

Vì một trường đại học có số lượng sinh viên theo môn học và số lượng môn học mà sinh viên đăng kí học có thể có nhiều chương trình đào tạo và mỗi chương trình đạo tạo sẽ có những tập hợp các môn học khác nhau, nghiên cứu này chỉ tập chung vào các sinh viên thuộc chương trình đại trà và chương trình chất lượng cao vì đây là hai chương trình có số lượng sinh viên lớn nhất và có cùng danh sách các môn học. Các sinh viên học các trương trình khác như chương trình tiên tiến, chương trình liên kết,. . . sẽ được loại bỏ để tránh nhiễu.

2.3 Outlier

Hình 4 cho thấy dữ liệu tồn tại outlier. Để loại bỏ những điểm này, những môn học với ít hơn 5 lượt đăng kí sẽ được loại bỏ. Cuối cùng, bộ dữ liệu còn lại 6571 dòng với 2500 sinh viên và 110 môn học.



Hình 4. Dữ liệu trước và sau khi tiền xử lý

Các thuộc tính được dùng để xây dựng đồ thị và phân tích dữ liệu được mô tả chi tiết trong bảng 1

STT	Thuộc tính	Mô tả
1	MSSV	MSSV của sinh viên
2	Ma_MH	Mã môn học
3	2_char_Ma_MH	2 kí tự đầu của mã môn học
4	Nganh_Hoc	Ngành học của sinh viên
5	2_char_Nganh_Hoc	2 kí tự đầu của ngành học
6	Nganh_Hoc_mapping	Ngành học đã được ánh xạ theo 2 kí tự đầu của mã môn học
7	learned	1 nếu sinh viên đã học môn này, ngược lại là 0

Bảng 1. Bảng mô tả thuộc tính

CHƯƠNG 3: XÂY DỰNG ĐỒ THỊ

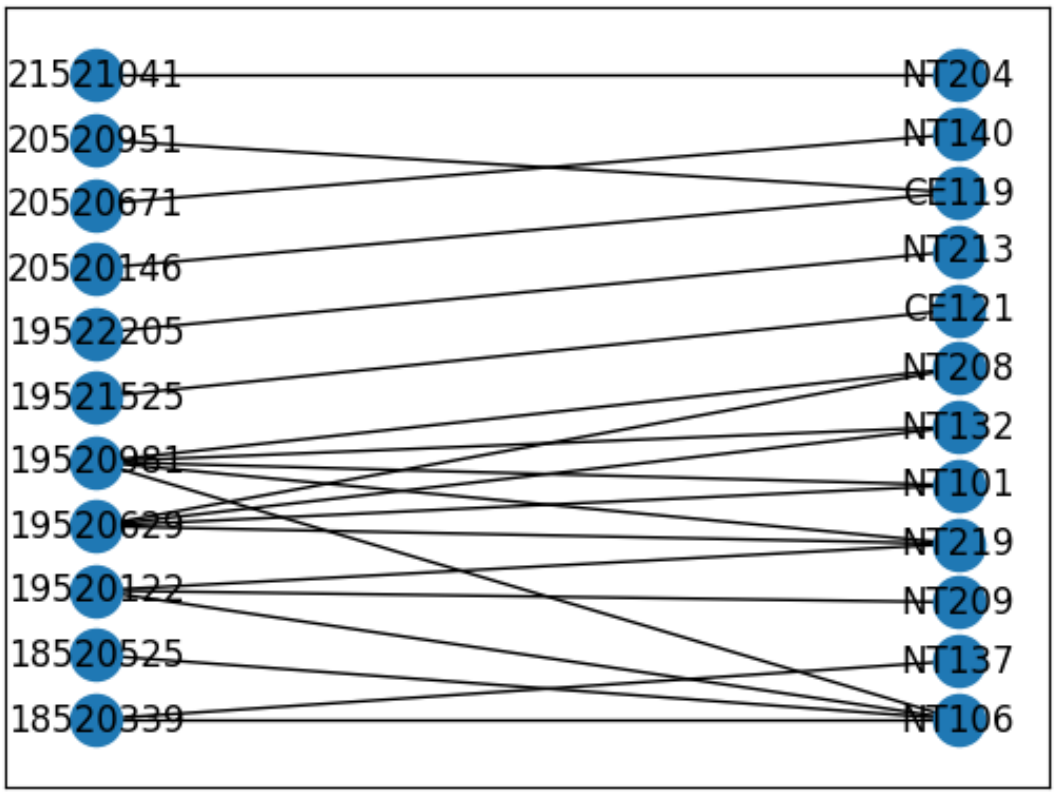
3.1 Đồ thị 2 phía

Node: Sinh viên hoặc môn học
Edge: Biểu diễn cho việc sinh viên học môn môn học

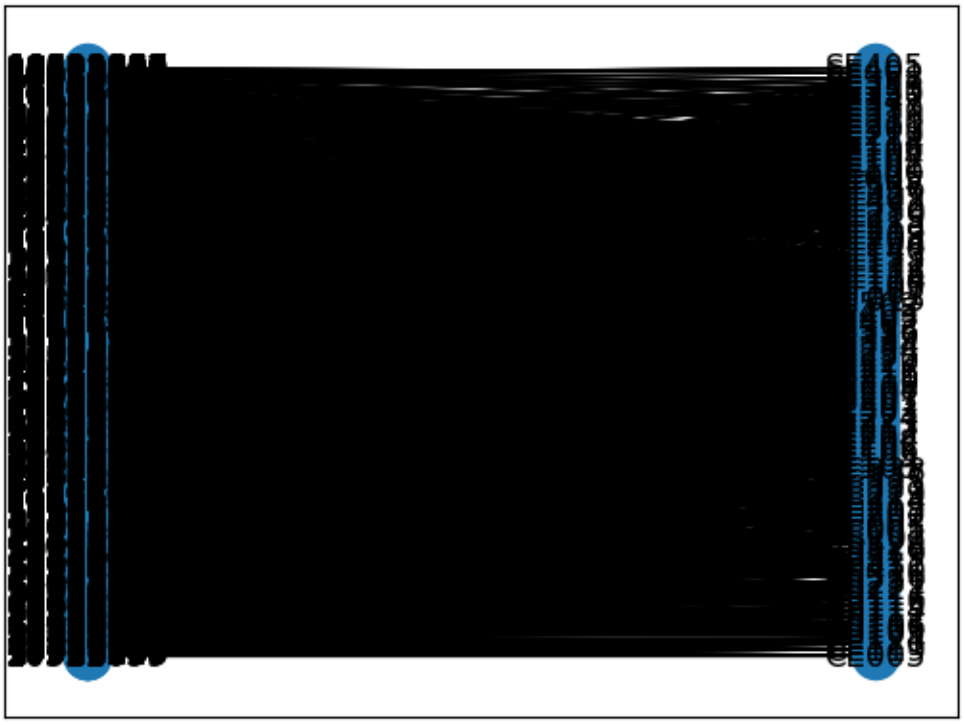
```
B=nx.Graph()
SinhVien = df_graph['MSSV']
MonHoc = df_graph['Ma_MH']
for index, row in df_graph.iterrows():
  B.add_edge(row['MSSV'],row['Ma_MH'],weight=1)
B.add_nodes_from(MonHoc, bipartite = 0)
B.add_nodes_from(SinhVien, bipartite = 1)
```

```
pos=nx.spring_layout(B)
nx.draw_networkx(B,pos=nx.drawing.layout.bipartite_layout(B, SinhVien))
```

Hình 5. Code tạo đồ thị 2 phía



Hình 6. Một phần đồ thị 2 phía biểu diễn tương tác giữa sinh viên và môn học



Hình 7. Toàn bộ đồ thị 2 phía

Nhìn vào đồ thị có thể thấy một môn học có nhiều sinh viên đăng kí và một sinh viên cũng học nhiều môn học.

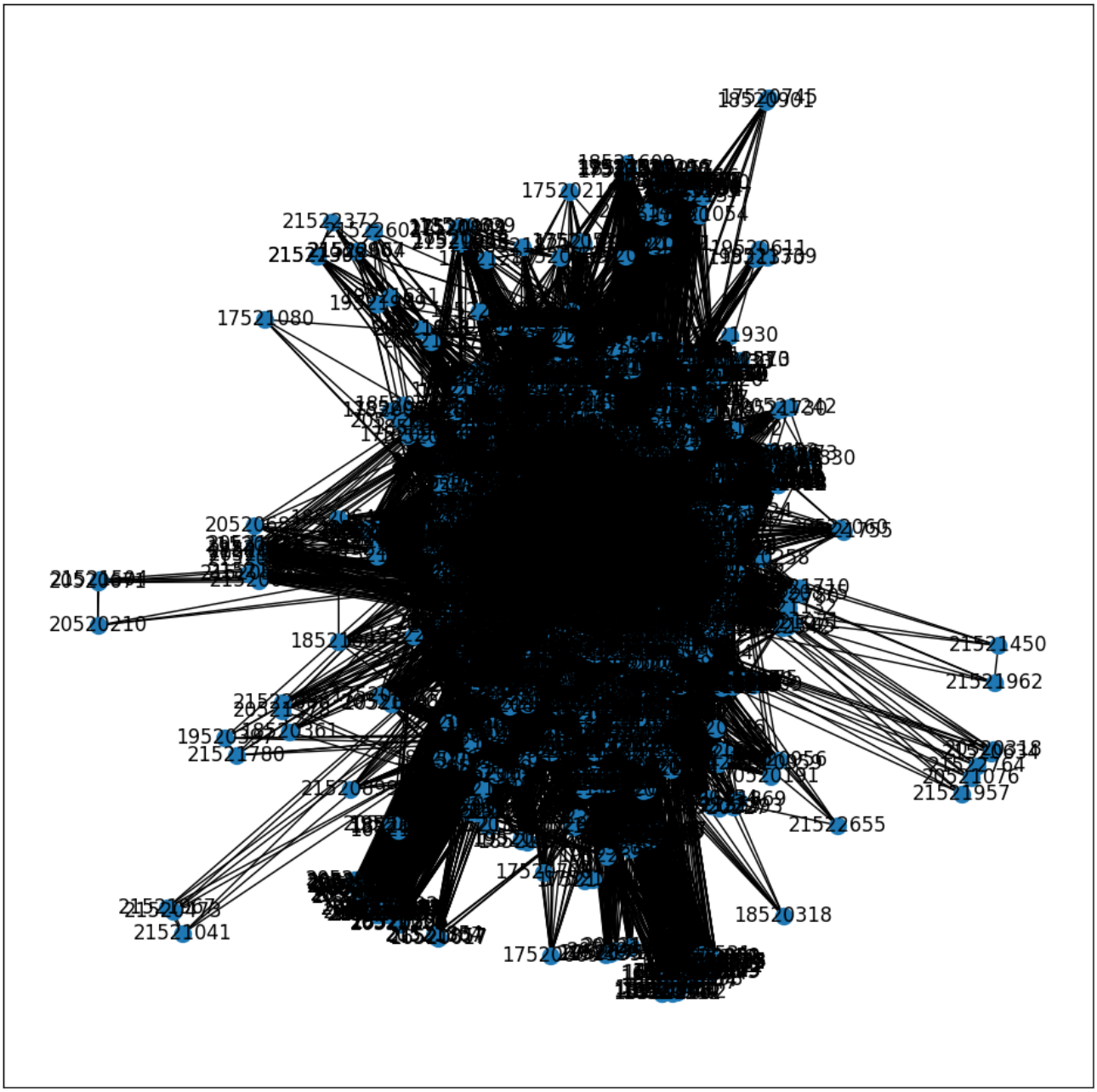
3.2 Đồ thị 1 phía

Node: Sinh viên
Edge: Biểu diễn cho việc 2 sinh viên cùng học một môn học
Weight: Trọng số là số lượng môn học mà 2 sinh viên cùng học

```
G=bipartite.weighted_projected_graph(B, SinhVien.unique())
```

```
plt.figure(figsize=(12,12))
layout = nx.spring_layout(G)

nx.draw_networkx_nodes(G, layout, nodelist=SinhVien.unique(),
                       node_size=100,)
nx.draw_networkx_edges(G,layout)
node_lables = dict(zip(SinhVien.unique(),SinhVien.unique()))
nx.draw_networkx_labels(G, layout, labels=node_lables)
plt.show()
```

Hình 8. Code tạo đồ thị 1 phía



Hình 9. Đồ thị 1 phía

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

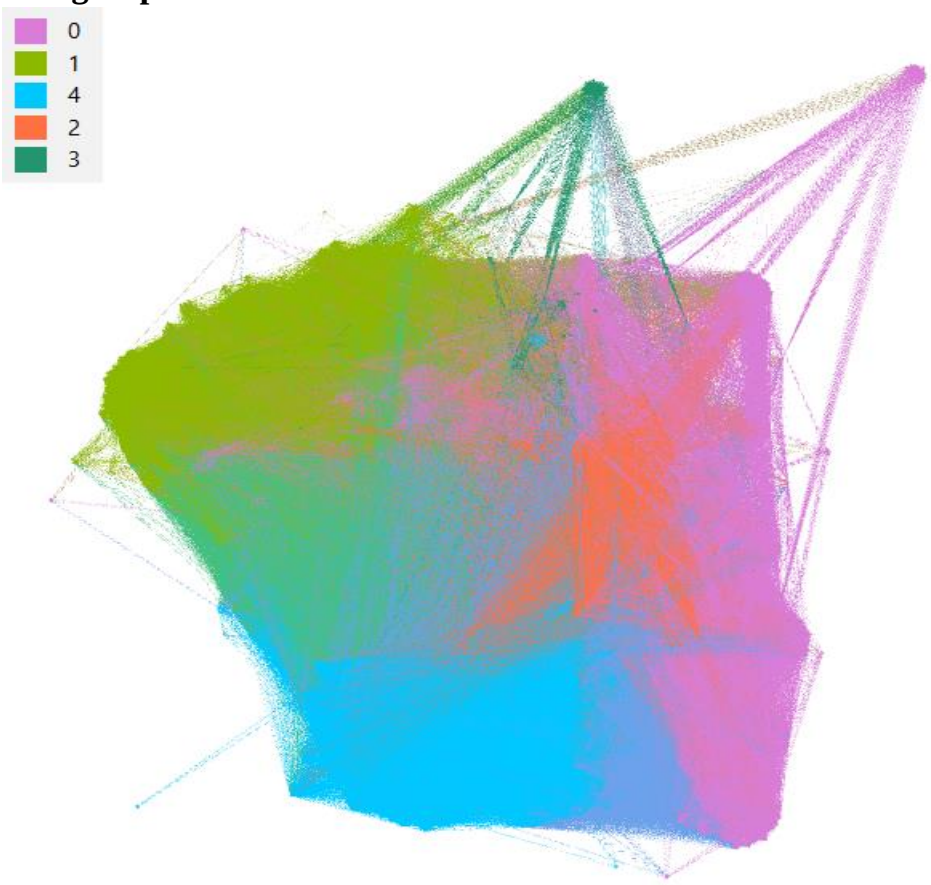
```
labels =nx.get_edge_attributes(G, 'weight')
source = []
target = []
weight = []
for key, value in labels.items():
  source.append(key[0])
  target.append(key[1])
  weight.append(value)

df2 = pd.DataFrame({'source': source,'target':target,'weight':weight})
df2.to_csv('gephi_weighted.csv', index=False)
```

Hình 10. Code lưu dữ liệu để thực hiện trên Gephi

CHƯƠNG 4: THUẬT TOÁN PHÁT HIỆN CỘNG ĐỒNG

4.1 Louvain

Thực hiện bằng Gephi



Hình 11. Kết quả phân cụm sử dụng louvain trên Gephi

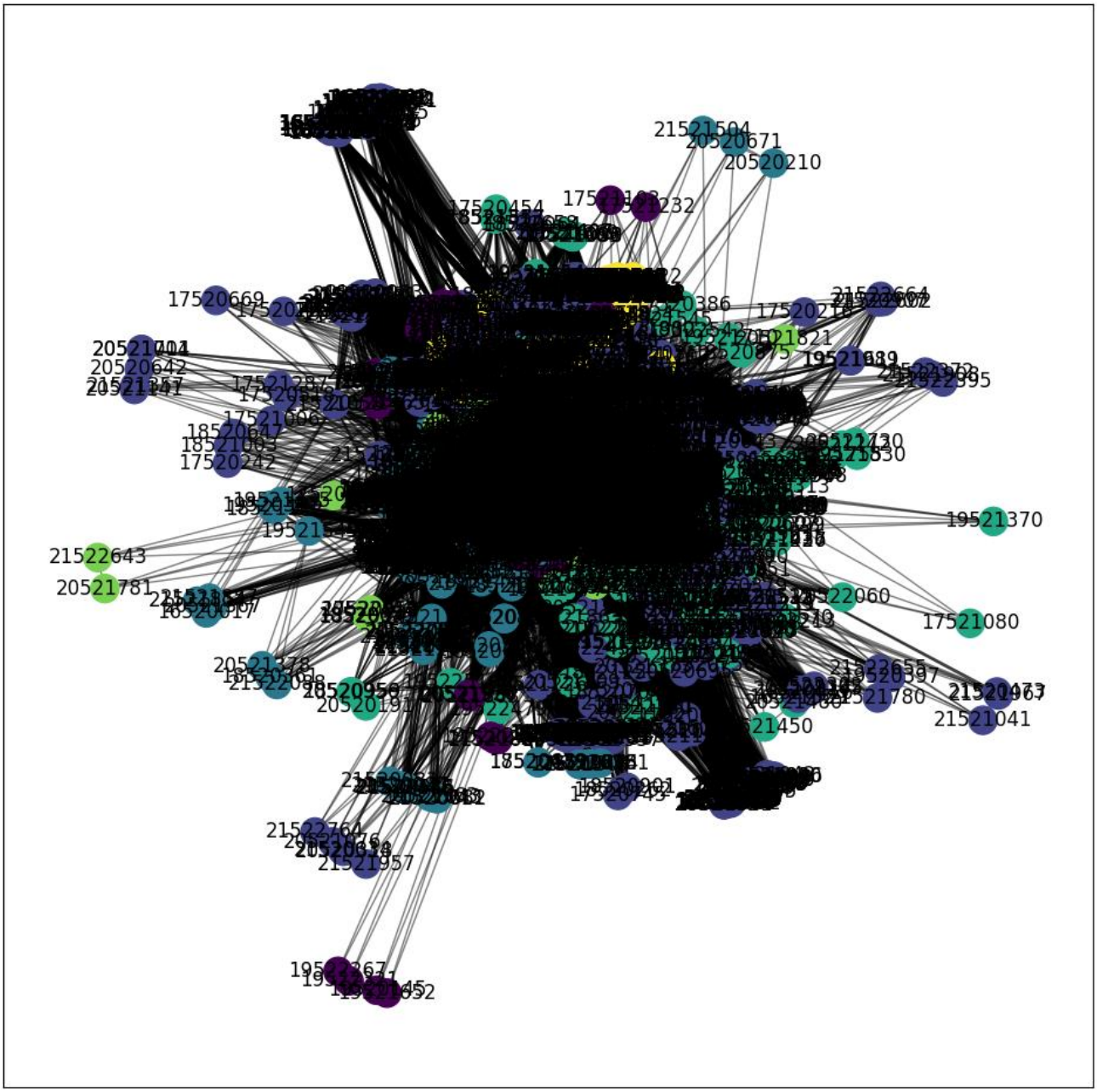
Kết quả thực thi thuật toán louvain bằng Gephi cho ra 5 cụm

Thực hiện bằng Python

```
partition = community_louvain.best_partition(G)
```

```
plt.figure(figsize=(12,12))

partition = community_louvain.best_partition(G)
pos=nx.spring_layout(G)
cmap=cm.get_cmap('viridis',max(partition.values()) + 1)
nx.draw_networkx_nodes(G, pos, partition.keys(), cmap=cmap, node_color=list(partition.values()))
nx.draw_networkx_edges(G, pos, alpha=0.5)
nx.draw_networkx_labels(G, pos)
plt.show()
```

Hình 12. Code phân cụm bằng louvain



Hình 13. Kết quả phân cụm băng louvain

Kết quả thực hiện bằng python cũng cho ra 5 cụm tương tự như Gephi

```
df_cluster = pd.DataFrame.from_dict(partition.items())
df_cluster.rename({0:'MSSV',1:'cluster'}, axis=1,inplace=True,)
df_cluster = df_cluster.sort_values(by='cluster')
df_cluster[['cluster']].value_counts()
```

```
cluster
1          721
4          649
3          416
2          403
0          311
```

4.2 Thuật toán Modularity

```
c = nx.community.greedy_modularity_communities(G)
```

```
df_modularity = pd.DataFrame([(node, i) for i, community in enumerate(c, 0) for node in community],
                             columns=['MSSV', 'cluster'])
```

Hình 14. Code phân cụm bằng modularity

```
df_modularity[['cluster']].value_counts()
```

cluster	
0	1074
1	950
2	338
3	47
4	43
5	23
6	10
7	8
8	7

Hình 15. Kết quả phân cụm bằng modularity

Thuật toán Modularity cho ra 9 cụm, tuy nhiên số lượng các cụm chênh lệch rất lớn. Các node đa số thuộc về cụm 0 và cụm 1.

4.3 Thuật toán Node2Vec + T-SNE + Kmeans

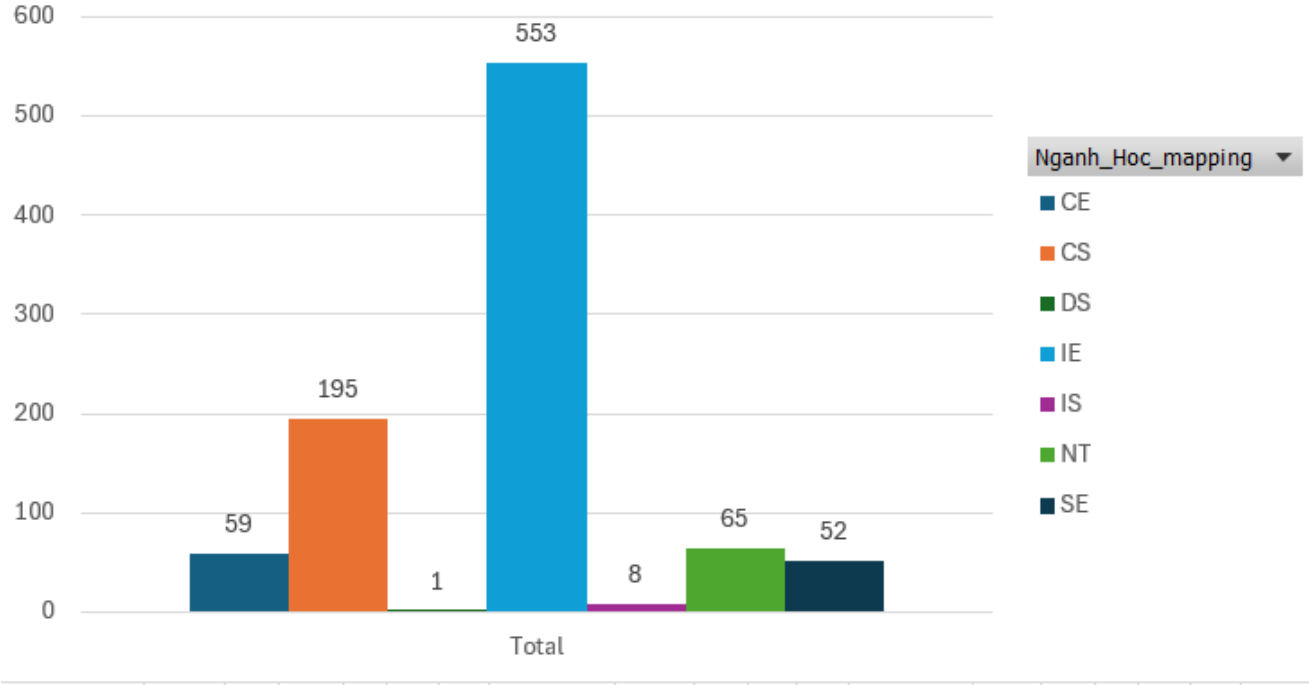
```
plt.figure(figsize=(8, 8))
plt.scatter(embeddings_2d[:, 0], embeddings_2d[:, 1], c=kmeans_2d.labels_, cmap='viridis')
plt.title('T-SNE Visualization of Node2Vec Embeddings with K-Means Clustering')
plt.legend()
plt.show()
```

Hình 16. Code vẽ kết quả phân cụm bằng K-means

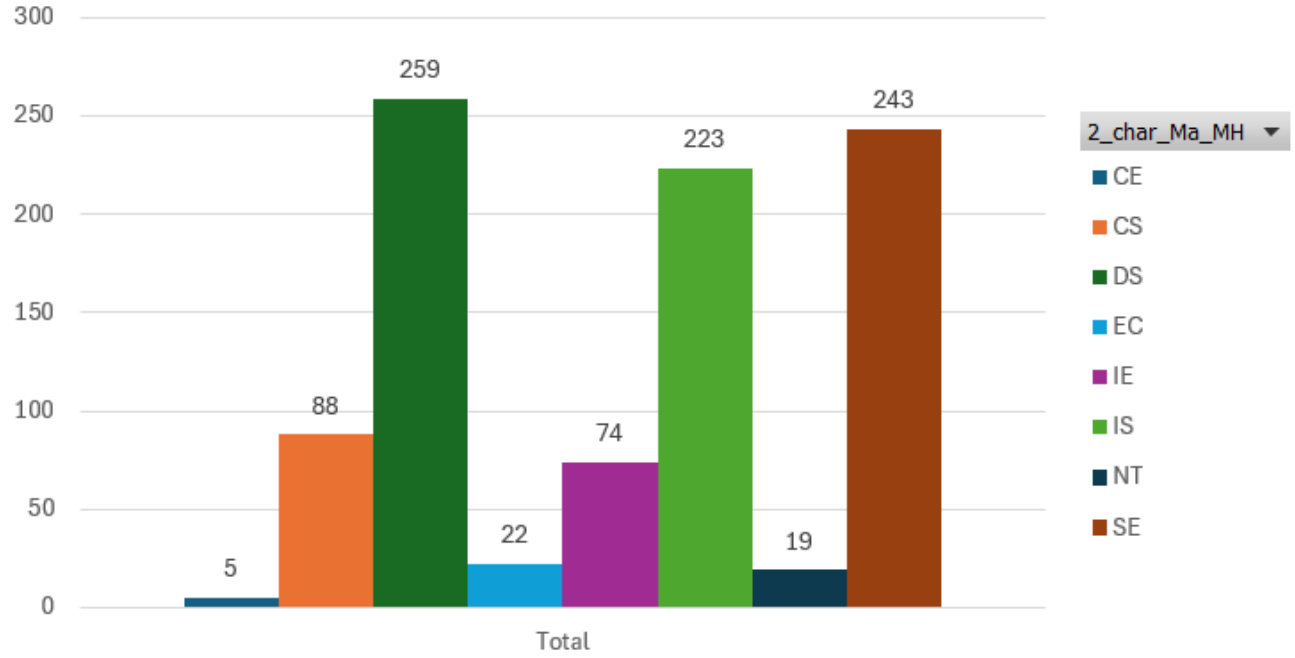
Hình 17. Kết quả phân cụm bằng K-means

CHƯƠNG 5: PHÂN TÍCH CỤM CỦA THUẬT TOÁN LOUVAIN

Cụm thứ 0:



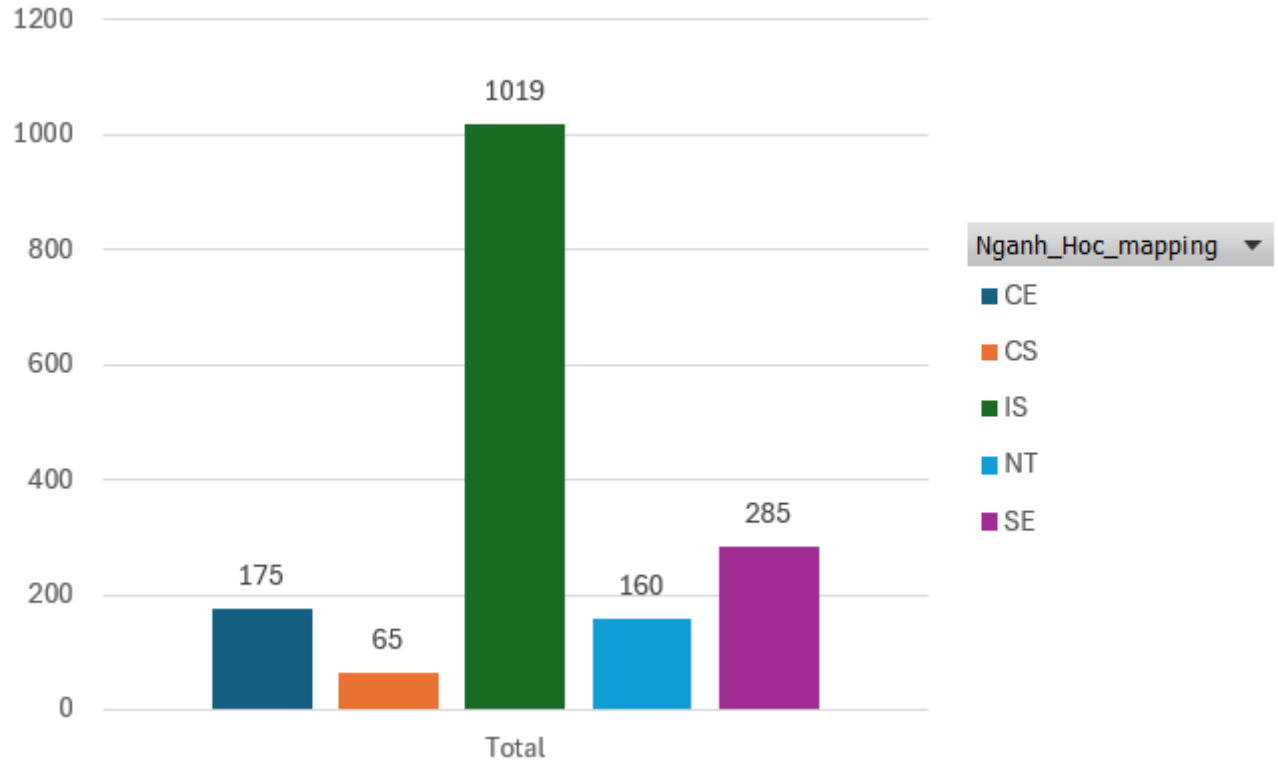
Hình 18. Thống kê số lượng sinh viên theo chuyên ngành của cụm 0



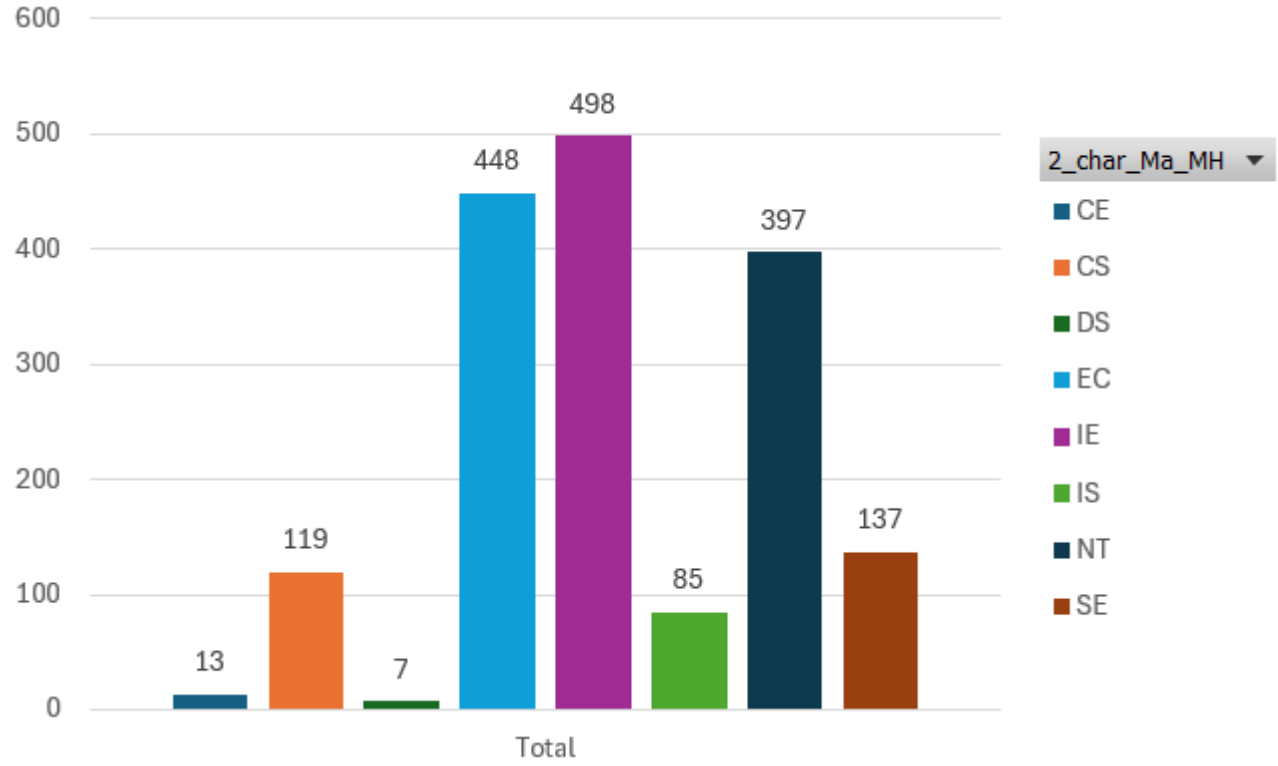
Hình 19. Thống kê số lượng sinh viên theo môn tự chọn đăng kí của cụm 0

Ý nghĩa: Ở cluster 0, sinh viên ngành Công nghệ thông tin (IE)chiếm đa số (553) những sinh viên này thường đăng kí học những môn tự chọn thuộc ngành Khoa học dữ liệu, Kỹ thuật phần mềm và Hệ thống thông tin.

Cụm thứ 1:



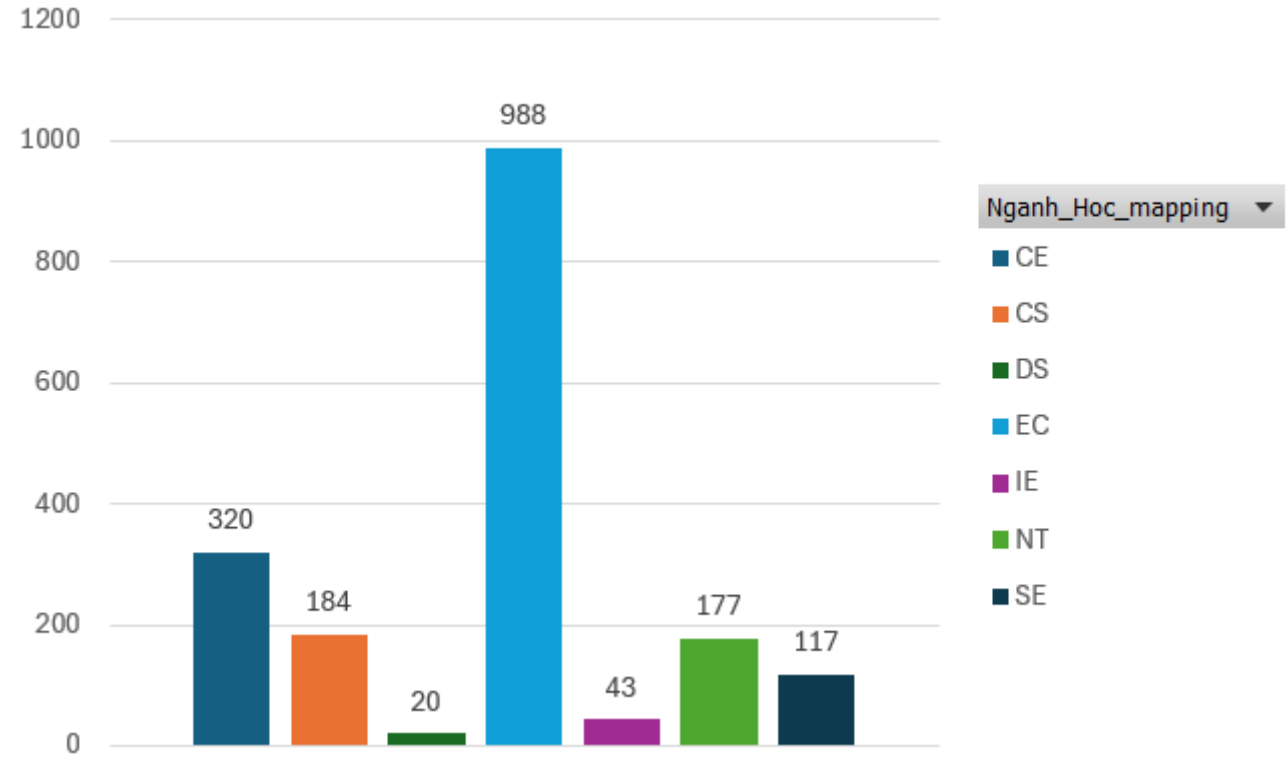
Hình 20. Thống kê số lượng sinh viên theo chuyên ngành của cụm 1



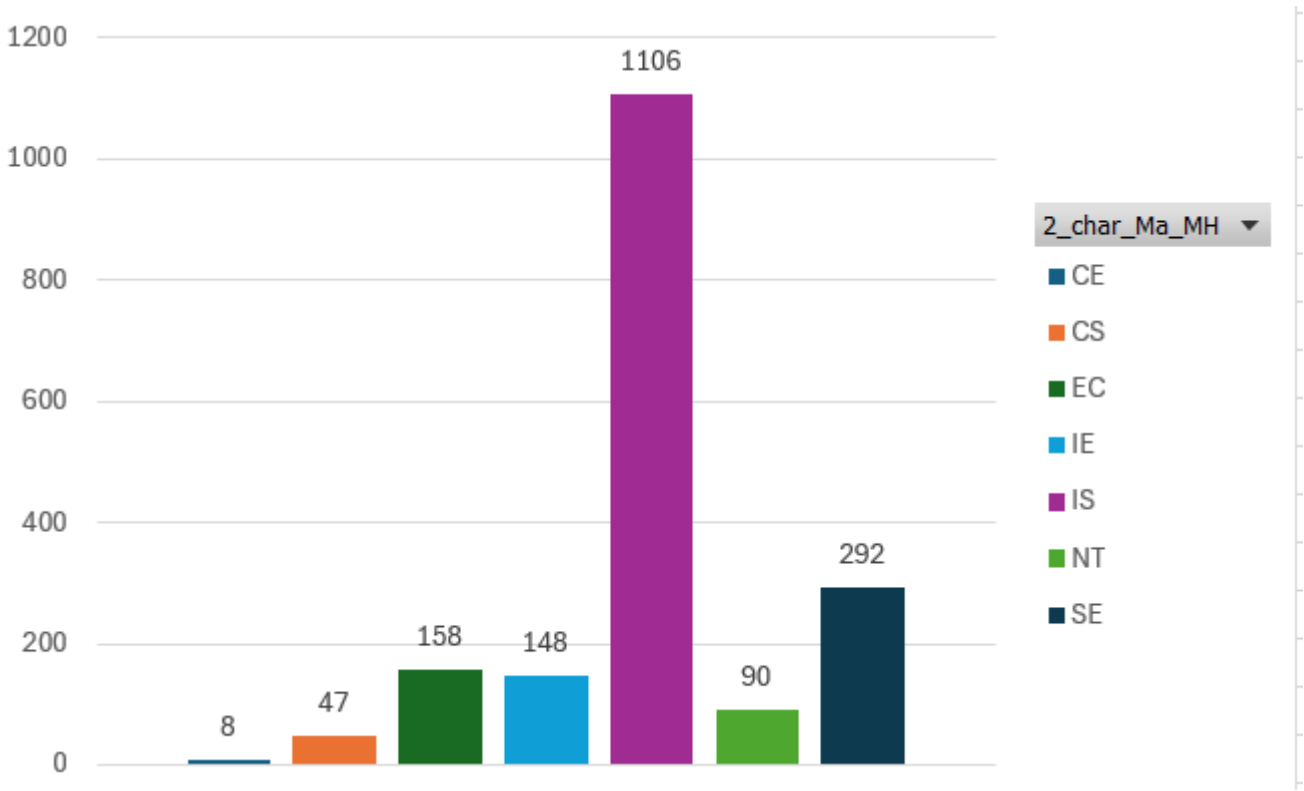
Hình 21. Thống kê số lượng sinh viên theo môn tự chọn đăng kí của cụm 1

Ý nghĩa: Ở cluster 1, sinh viên ngành Hệ thống thông tin (IS) chiếm đa số (1019) những sinh viên này thường đăng kí học những môn tự chọn thuộc ngành Công nghệ thông tin, Thương mại điện tử và Mạng máy tính.

Cụm thứ 2:



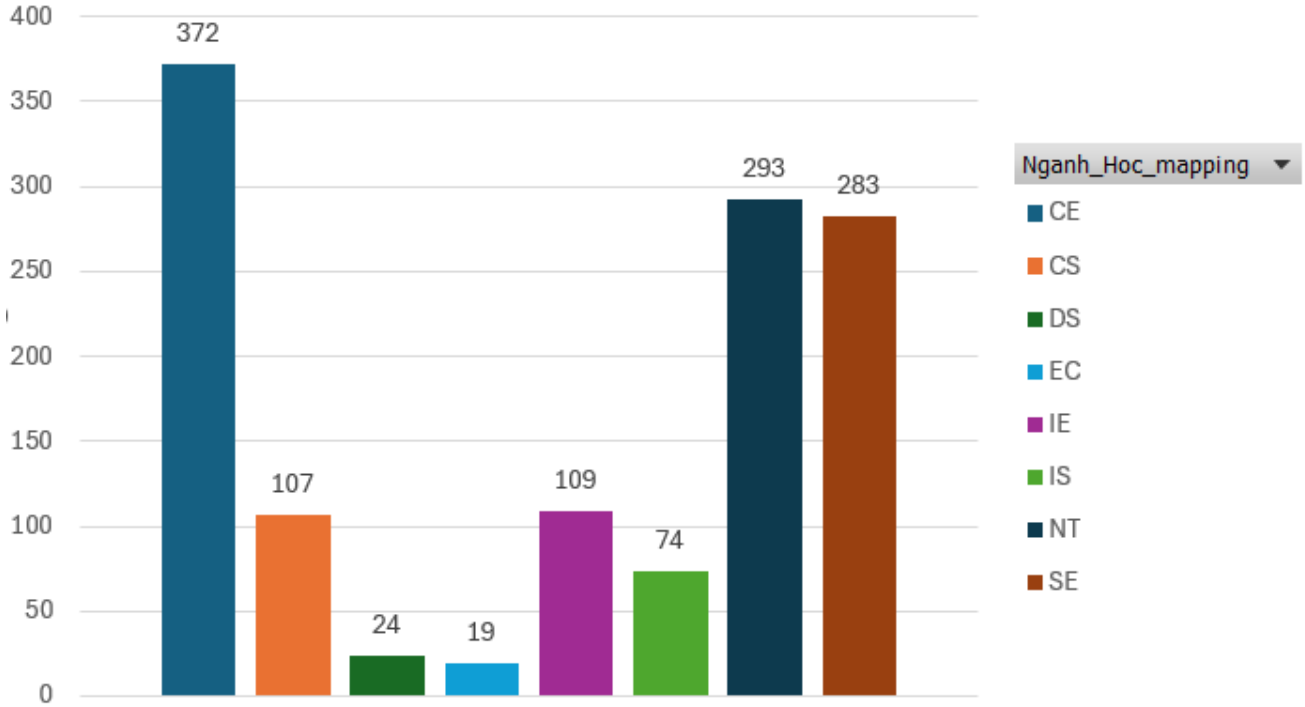
Hình 22. Thống kê số lượng sinh viên theo chuyên ngành của cụm 2



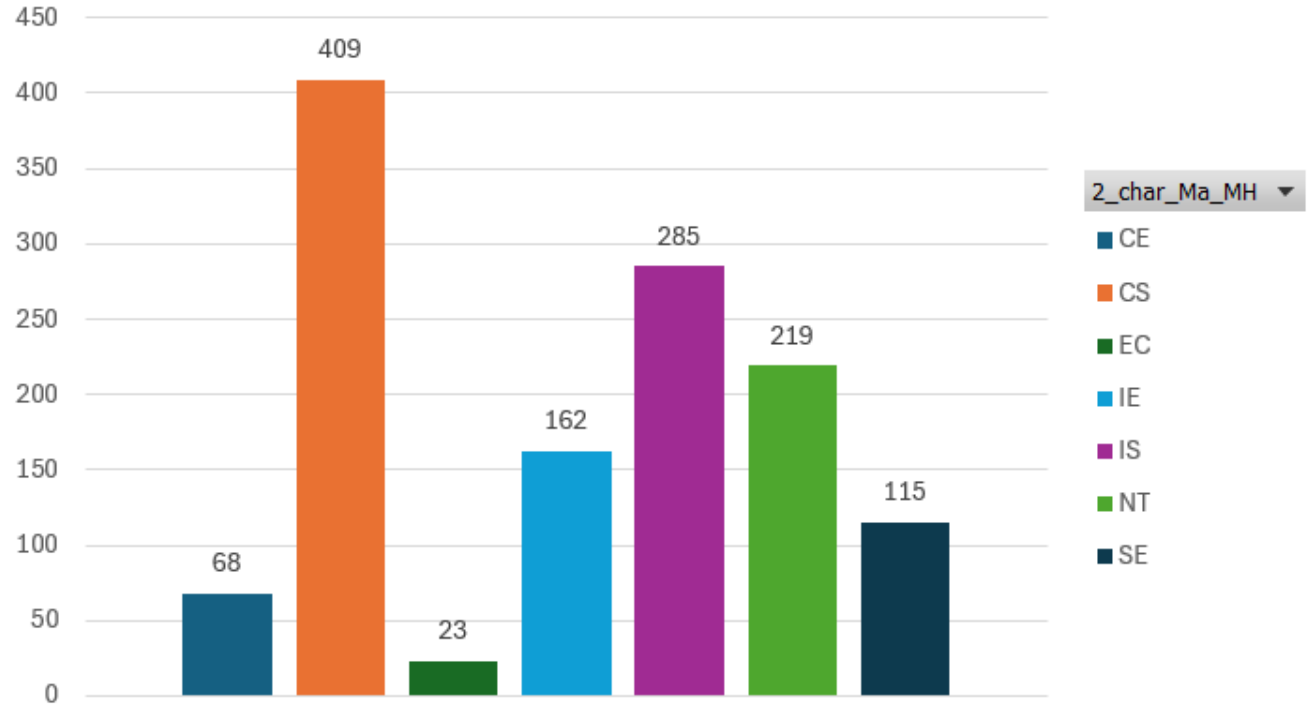
Hình 23. Thống kê số lượng sinh viên theo môn tự chọn đăng kí của cụm 2

Ý nghĩa: Ở cluster 2, sinh viên ngành Thương mại điện tử (EC) chiếm đa số (988) những sinh viên này thường đăng kí học những môn tự chọn thuộc ngành Hệ thông thông tin.

Cụm thứ 3:



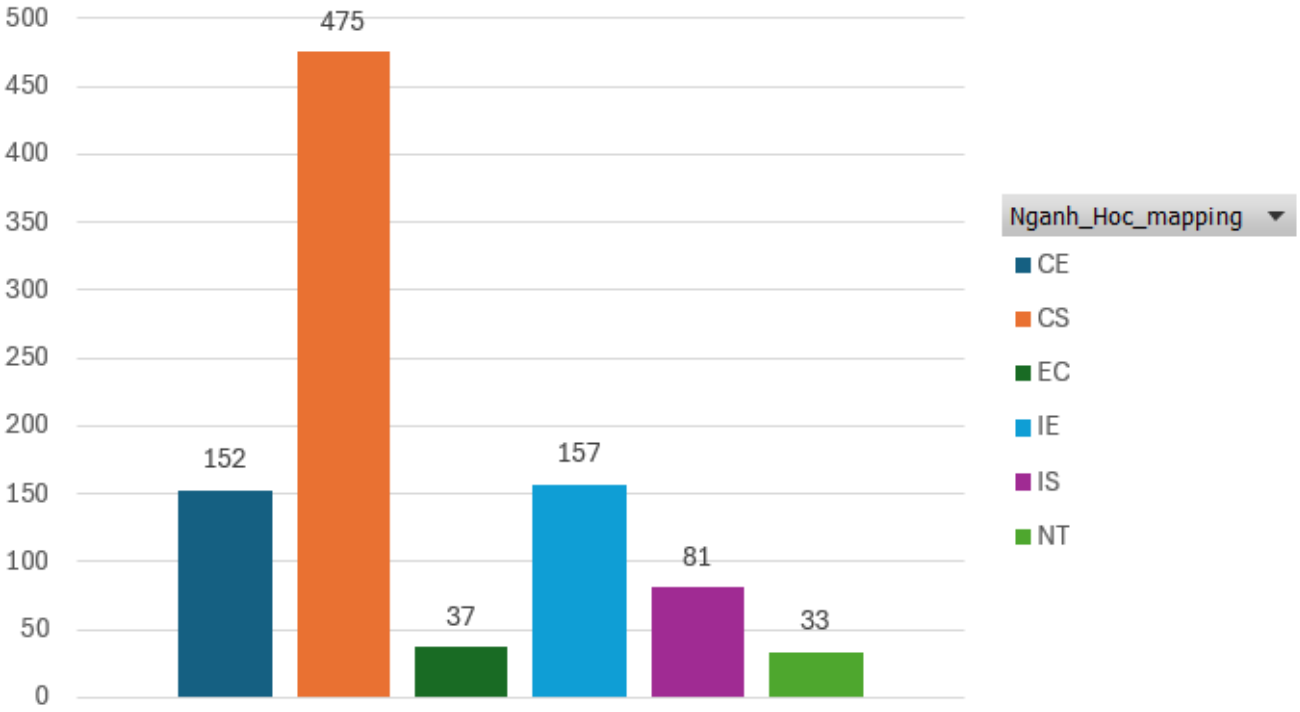
Hình 24. Thống kê số lượng sinh viên theo chuyên ngành của cụm 3



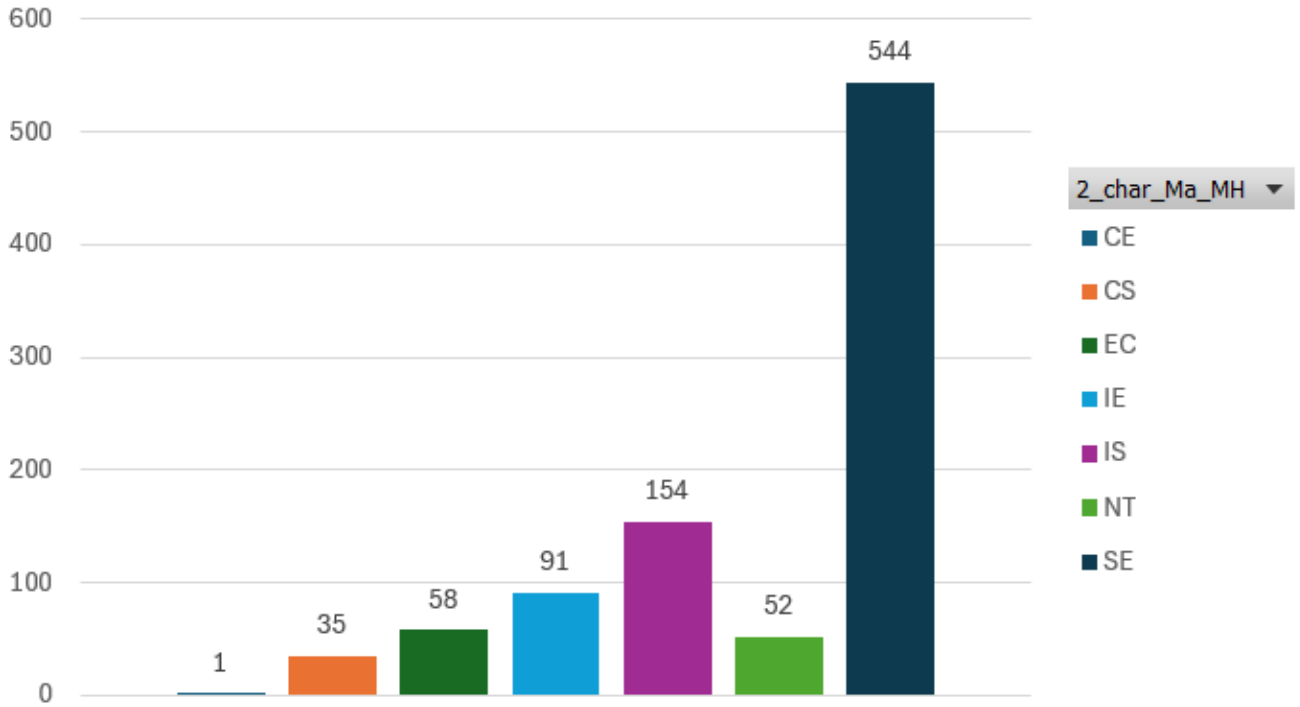
Hình 25. Thống kê số lượng sinh viên theo môn tự chọn đăng kí của cụm 3

Ý nghĩa: Ở cluster 3, sinh viên ngành Kỹ thuật máy tính (CE) chiếm đa số (372) những sinh viên này thường đăng kí học những môn tự chọn thuộc ngành Khoa học máy tính.

Cụm thứ 4:



Hình 26. Thống kê số lượng sinh viên theo chuyên ngành của cụm 4



Hình 27. Thống kê số lượng sinh viên theo môn tự chọn đăng kí của cụm 4

Ý nghĩa: Ở cluster 4, sinh viên ngành Khoa học máy tính chiếm đa số (475)
những sinh viên này thường đăng kí học những môn tự chọn thuộc ngành Kỹ thuật phần mềm

Kết luận: *Theo thuật toán louvain, các sinh viên được chia thành 5 cụm. Trong 5 cụm này, mỗi cụm đều có sinh viên của 1 ngành chiếm đa số. Các sinh viên này có xu hướng học các môn của các ngành như Công nghệ thông tin, Thương mại điện tử và Mạng máy tính, Hệ thông thông tin, Khoa học máy tính, Kỹ thuật phần mềm và 2 ngành còn lại là Khoa học dữ*

liệu và Kỹ thuật máy tính thường có ít sinh viên ngành khác học hơn. Vậy có 6 ngành có nhiều sinh viên các ngành khác đăng ký và 2 ngành có ít sinh viên ngành khác đăng ký.