



Nhóm 13

Khoa học dữ liệu

Phân Cụm Thị Trường Cầu Thủ Bóng Đá

Sinh viên thực hiện:

- Lê Phước Duy
- Nguyễn Phan Bảo Lộc
- Lê Xuân Tiến Nhật

Giáo viên hướng dẫn: TS. Ninh Khánh Duy



Nội dung

• • • •

1.

Mã hóa đặc trưng
danh mục

2.

Trích xuất
đặc trưng

3.

Mô hình phân cụm

4.

Khảo sát,
đánh giá

• • • •

1. Mã hóa đặc trưng danh mục

- Vì hầu hết các mô hình học máy chỉ chấp nhận các biến số nên cần xử lý trước các biến phân loại. Cần chuyển đổi các biến phân loại thành số sao cho mô hình có thể hiểu và trích xuất thông tin có giá trị.
- Có 2 loại dữ liệu danh mục: dữ liệu thứ tự và dữ liệu danh mục
- Các kỹ thuật mã hóa đặc trưng danh mục: Label Encoding, Ordinal Encoding, One-Hot Encoding, Dummy Encoding, Frequency Encoding, Target Encoding, Hash Encoding
- Với dữ liệu “Position” và “Preferred_foot” là kiểu dữ liệu danh mục danh nghĩa  Label Encoding

Label Encoding:

- Được sử dụng để chuyển đổi các giá trị của biến danh mục thành các số nguyên
- Label Encoding không tạo ra mối quan hệ thứ bậc giữa các giá trị và nó thường sử dụng cho các biến mà không có mối quan hệ thứ bậc

```
from sklearn.preprocessing import LabelEncoder  
label_position_encoder = LabelEncoder()  
players['position'] = label_position_encoder.fit_transform(players['position'])
```

2. Trích xuất đặc trưng

2.1 Biến đổi đặc trưng

a) Chuẩn hóa (standardization)

- Kỹ thuật chuẩn hóa được áp dụng đối với những biến có phân phối chuẩn. Biến được biến đổi theo kì vọng và độ lệch chuẩn như sau:

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma(\mathbf{x})}$$

- Các biến sau khi được chuẩn hóa sẽ có cùng một dạng phân phối chuẩn hóa với trung bình bằng 0 và phương sai bằng 1.

 Nhờ đó quá trình huấn luyện sẽ trở nên ổn định và hội tụ tới nghiệm tối ưu nhanh hơn.

- Là một phép co trong không gian mà ở đó khoảng cách giữa 2 điểm bất kì luôn cùng một tỷ lệ so với không gian gốc.

2. Trích xuất đặc trưng

2.1 Biến đổi đặc trưng

b) Kĩ thuật scaling

- Kĩ thuật scaling thường áp dụng trên những biến không tuân theo phân phối chuẩn. Thông qua scaling, toàn bộ giá trị của biến sẽ được đưa về một miền giá trị bị giới hạn trong khoảng [0,1].
- Trong kĩ thuật scaling thì chúng ta có các phương pháp chính: Minmax Scaling, Unit Length, Robust Scaling

Minmax Scaling

- Biến được đưa về các range [0,1] theo công thức:

$$\mathbf{x}' = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$$

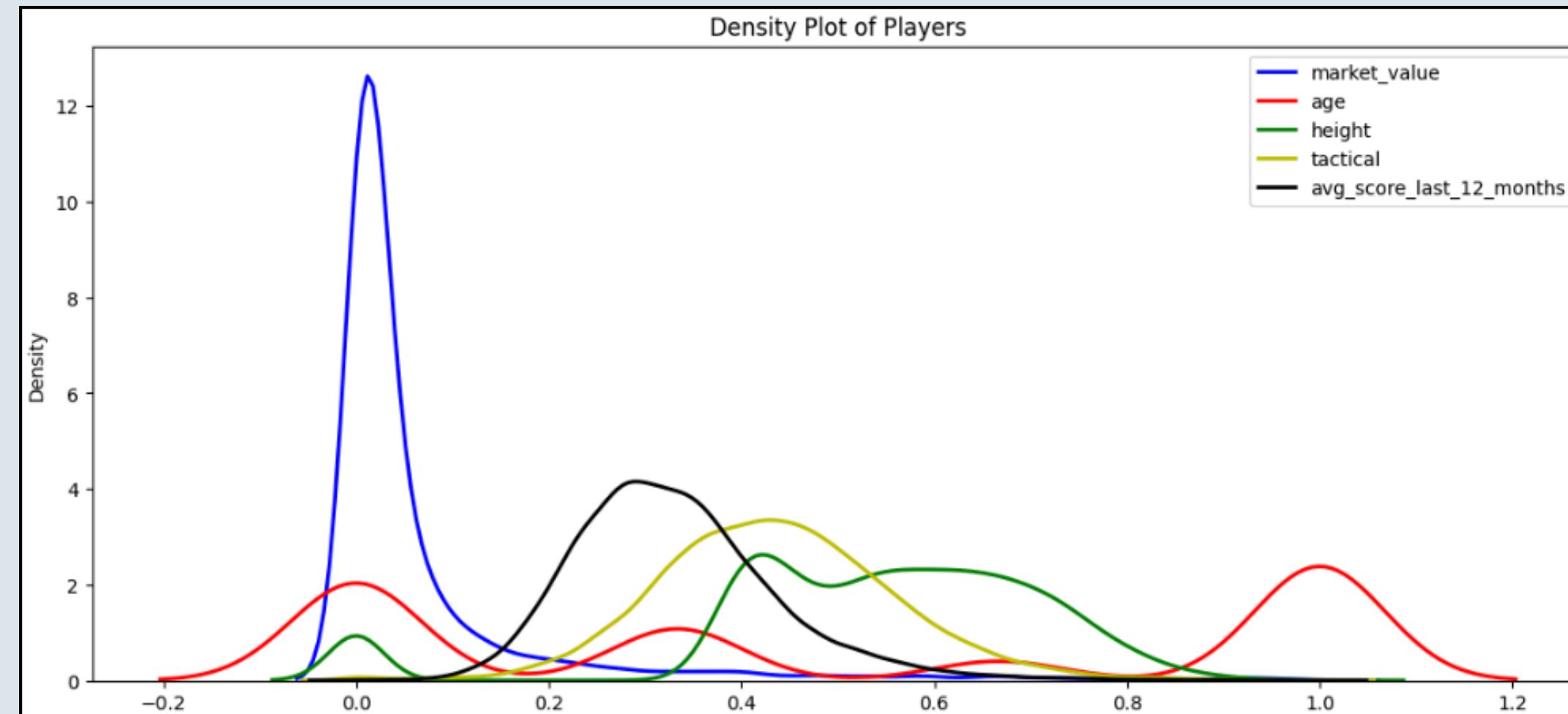
- Là một phép co trong không gian mà ở đó tỷ lệ khoảng cách giữa 2 điểm bất kì được bảo toàn so với khoảng cách của chúng trong không gian gốc.

```
from sklearn.preprocessing import MinMaxScaler
min_max = MinMaxScaler()
players = pd.DataFrame(min_max.fit_transform(players), columns=players.columns)
```

Kết quả biến đổi

	market_value	age	height	preferred_foot	position	attacking	technical	tactical	defending	creativity	saves	anticipation
0	0.184531	0.222222	0.66		0.5	0.333333	0.781609	0.617978	0.5000	0.358696	0.526882	0.0
1	0.061961	0.333333	0.46		1.0	0.333333	0.712644	0.539326	0.5375	0.347826	0.505376	0.0
2	0.029442	0.222222	0.56		1.0	0.333333	0.735632	0.573034	0.4250	0.315217	0.516129	0.0

ball_distribution	aerial	avg_score_last_12_months	appearances_last_12_months	point_of_tournament
0.0	0.0	0.438424	0.410714	1.0
0.0	0.0	0.300493	0.482143	1.0
0.0	0.0	0.270936	0.428571	1.0



2. Trích xuất đặc trưng

2.2 Lựa chọn đặc trưng

- Mô hình có quá nhiều trường dữ liệu cũng không thực sự tốt. Có thể gây ra các hạn chế:
 - Tăng chi phí tính toán.
 - Quá nhiều biến giải thích có thể dẫn tới quá khớp (overfitting).
 - Trong số các biến sẽ có những biến gây nhiễu và làm giảm chất lượng mô hình.
 - Rối loạn thông tin do không thể kiểm soát và hiểu hết các biến.

➡ cần phải có những phương pháp như giảm chiều dữ liệu hoặc lựa chọn biến quan trọng.

- Đối với việc lựa chọn biến quan trọng, có một số phương pháp lựa chọn sau:
 - Phương pháp thống kê
 - Sử dụng mô hình
 - Sử dụng Search

```
from sklearn.preprocessing import StandardScaler
std = StandardScaler()
players[['age', 'height', 'tactical']] = std.fit_transform(players[['age', 'height', 'tactical']])
```

2. Trích xuất đặc trưng

2.2 Lựa chọn đặc trưng

Phương pháp thống kê

- Phân tích các biến không biến động thì không có tác dụng gì trong việc phân loại
- Thông qua độ lớn phương sai của các biến numeric để loại bỏ những biến nếu nó nhỏ hơn một người nhất định.

```
from sklearn.feature_selection import VarianceThreshold

def variance_threshold_selector(data, threshold=0.01):
    selector = VarianceThreshold(threshold)
    selector.fit(data)
    return data[data.columns[selector.get_support(indices=True)]]
```

```
players = variance_threshold_selector(players)

players.head(3)
```

	market_value	age	height	preferred_foot	position	attacking	technical	tactical	defending	creativity	saves	anticipation	ball_distribution	aerial	avg_score_last_12_months	appearances_last_12_months	point_of_tournament	
0	0.184531	0.222222	0.66		0.5	0.333333	0.781609	0.617978	0.5000	0.358696	0.526882	0.0	0.0	0.0	0.0	0.438424	0.410714	1.0
1	0.061961	0.333333	0.46		1.0	0.333333	0.712644	0.539326	0.5375	0.347826	0.505376	0.0	0.0	0.0	0.0	0.300493	0.482143	1.0
2	0.029442	0.222222	0.56		1.0	0.333333	0.735632	0.573034	0.4250	0.315217	0.516129	0.0	0.0	0.0	0.0	0.270936	0.428571	1.0



Lựa chọn tất cả các biến đặc trưng, không loại bỏ biến nào.

3. Áp dụng mô hình phân cụm

3.1 Lựa chọn tham số

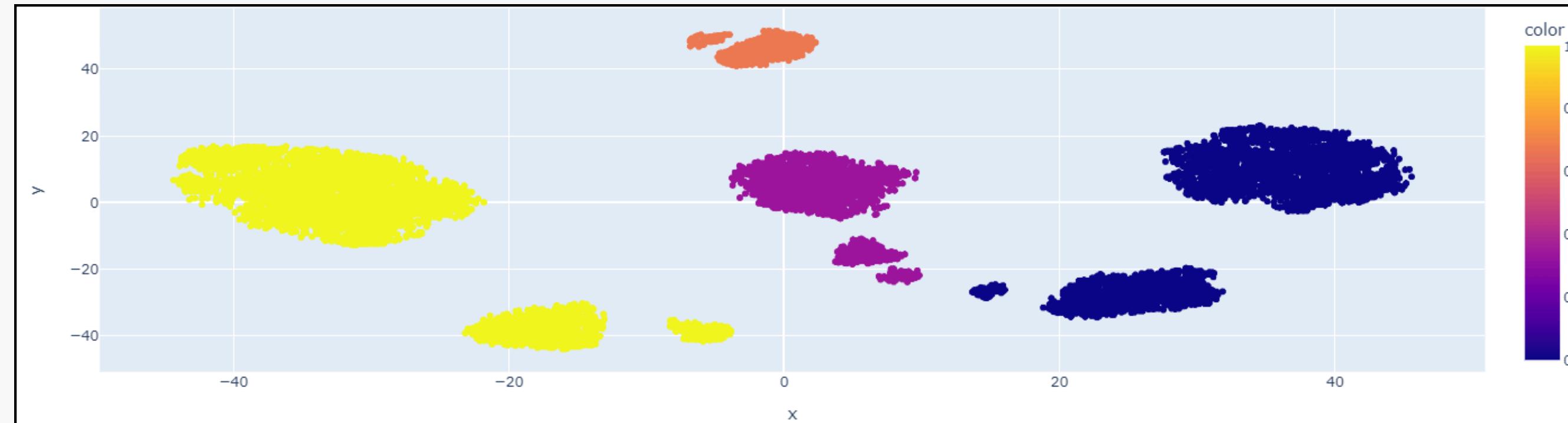
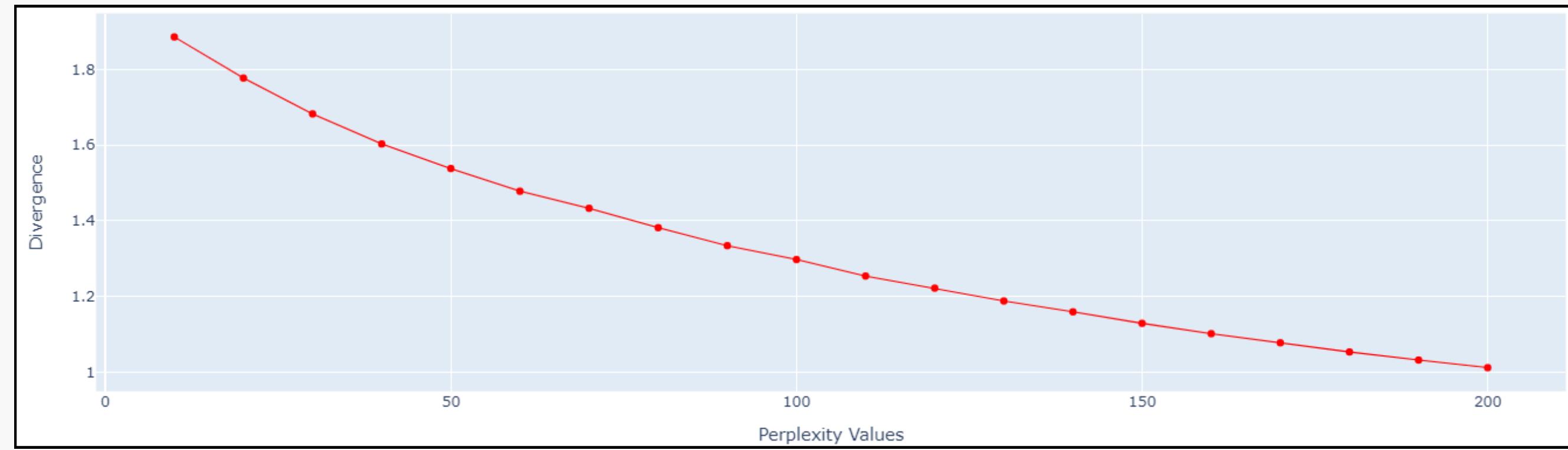
- Vì dữ liệu trước và sau khi phân cụm trước khi trực quan cần giảm số chiều dữ liệu bằng t-SNE
➡ Cần khảo sát các tham số quan trọng khi sử dụng t-SNE để giảm chiều và trực quan dữ liệu
- Các tham số cần chú ý khi sử dụng t-SNE:
 - n_components = 2 : giảm chiều dữ liệu về 2
 - perplexity : Độ phức tạp của phân phối xác suất được sử dụng để mô hình hóa sự tương đồng giữa các điểm dữ liệu (mặc định bằng 30)
 - random_state = 42 : Trạng thái ngẫu nhiên để đảm bảo tính nhất quán của kết quả.

```
perplexity = np.arange(10, 210, 10)
divergence = []

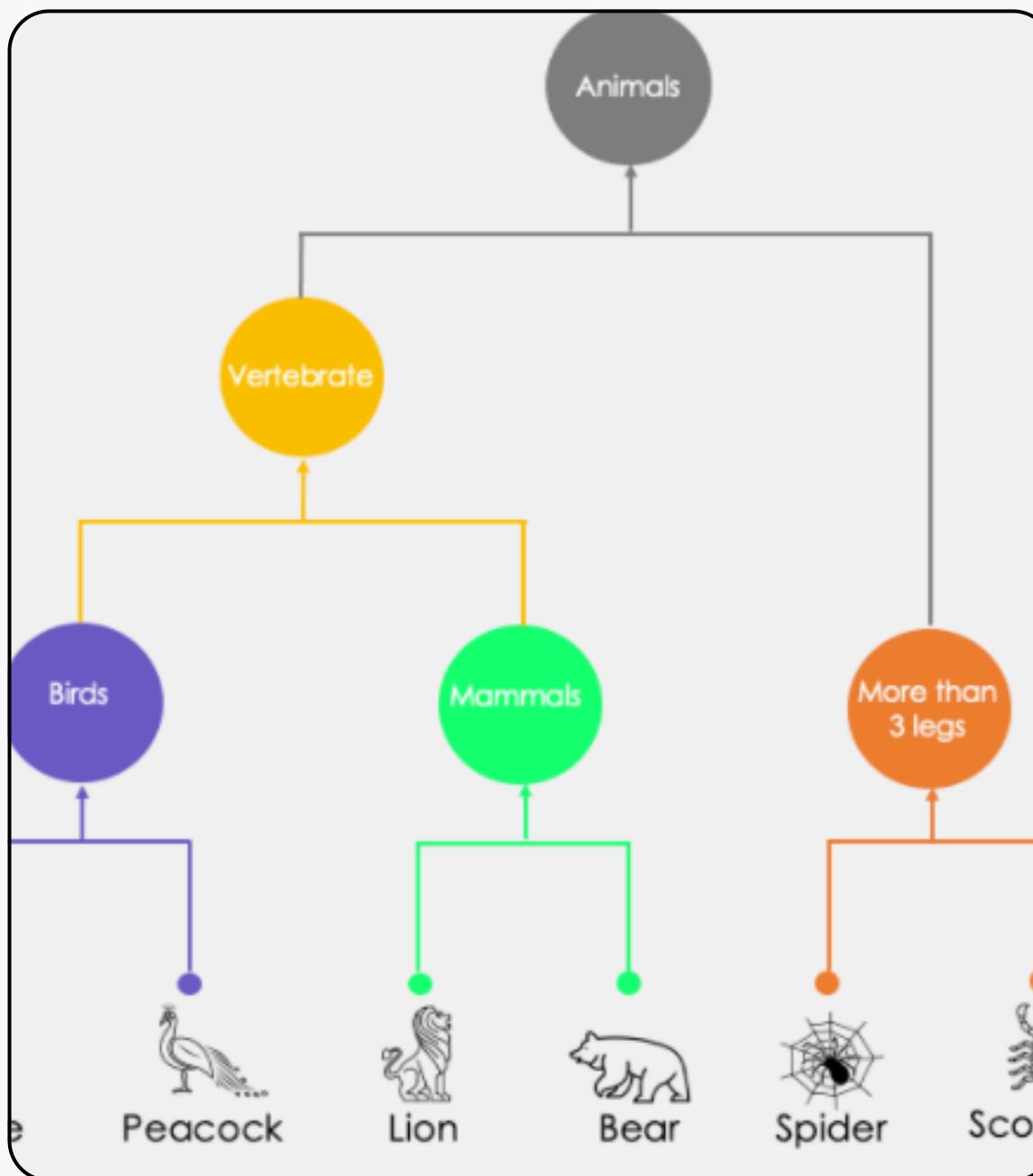
for i in perplexity:
    model = TSNE(n_components=2, perplexity=i)
    reduced = model.fit_transform(players)
    divergence.append(model.kl_divergence_)
fig = px.line(x=perplexity, y=divergence, markers=True)
fig.update_layout(xaxis_title="Perplexity Values", yaxis_title="Divergence")
fig.update_traces(line_color="red", line_width=1)
fig.show()
```

3. Áp dụng mô hình phân cụm

3.1 Lựa chọn tham số



Hierarchical clustering (phân cụm phân cấp)



Giới thiệu:

- Đây là một thuật toán phân tích cụm nhằm xây dựng hệ thống phân cấp các cụm trong dữ liệu. Nó hoạt động bằng cách lặp đi lặp lại việc sáp nhập hoặc chia tách các cụm dựa trên mức độ giống nhau giữa các điểm dữ liệu.

Cách hoạt động:

- Mỗi điểm dữ liệu là một cụm riêng biệt.
- Hợp nhất/ Phân chia cụm.

Ưu điểm:

- Khả năng khám phá: Khám phá cấu trúc ẩn trong dữ liệu mà không cần biết trước số lượng cụm.
- Dễ hình dung: Dendrogram giúp dễ dàng hình dung mối quan hệ phân cấp.
- Có thể áp dụng cho nhiều loại dữ liệu.

Nhược điểm:

- Độ phức tạp: Tốn thời gian tính toán cho các tập dữ liệu lớn.
- Khó khăn trong việc xác định số lượng cụm: Việc xác định số lượng cụm mong muốn từ cây dendrogram có thể khó khăn.

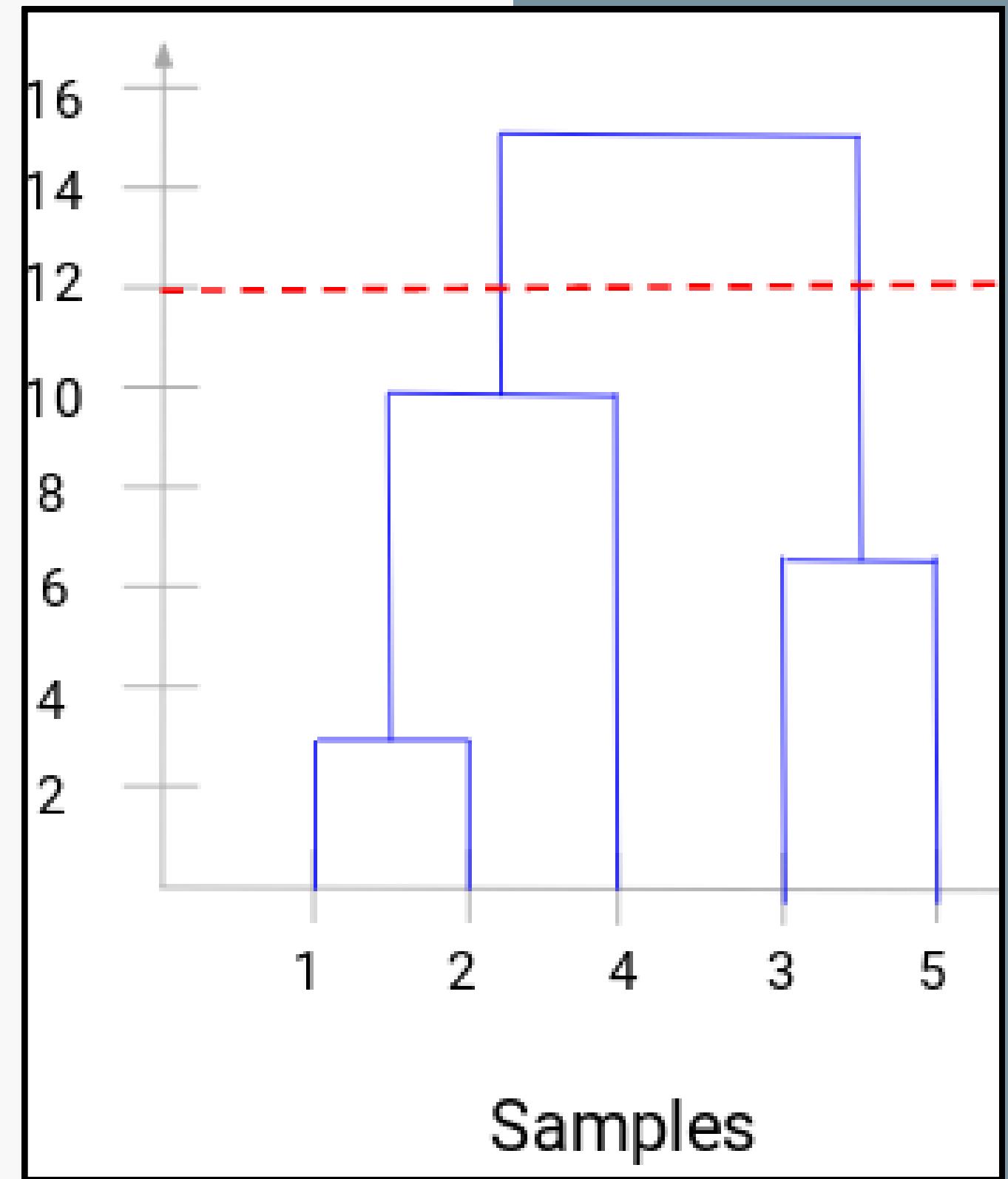
Dendrogram

Giới thiệu:

Dendrogram, hay còn gọi là biểu đồ cây, là một dạng biểu đồ được sử dụng để minh họa cấu trúc phân cấp của các cụm giúp người dùng dễ dàng hiểu và phân tích kết quả phân cụm.

Cấu trúc:

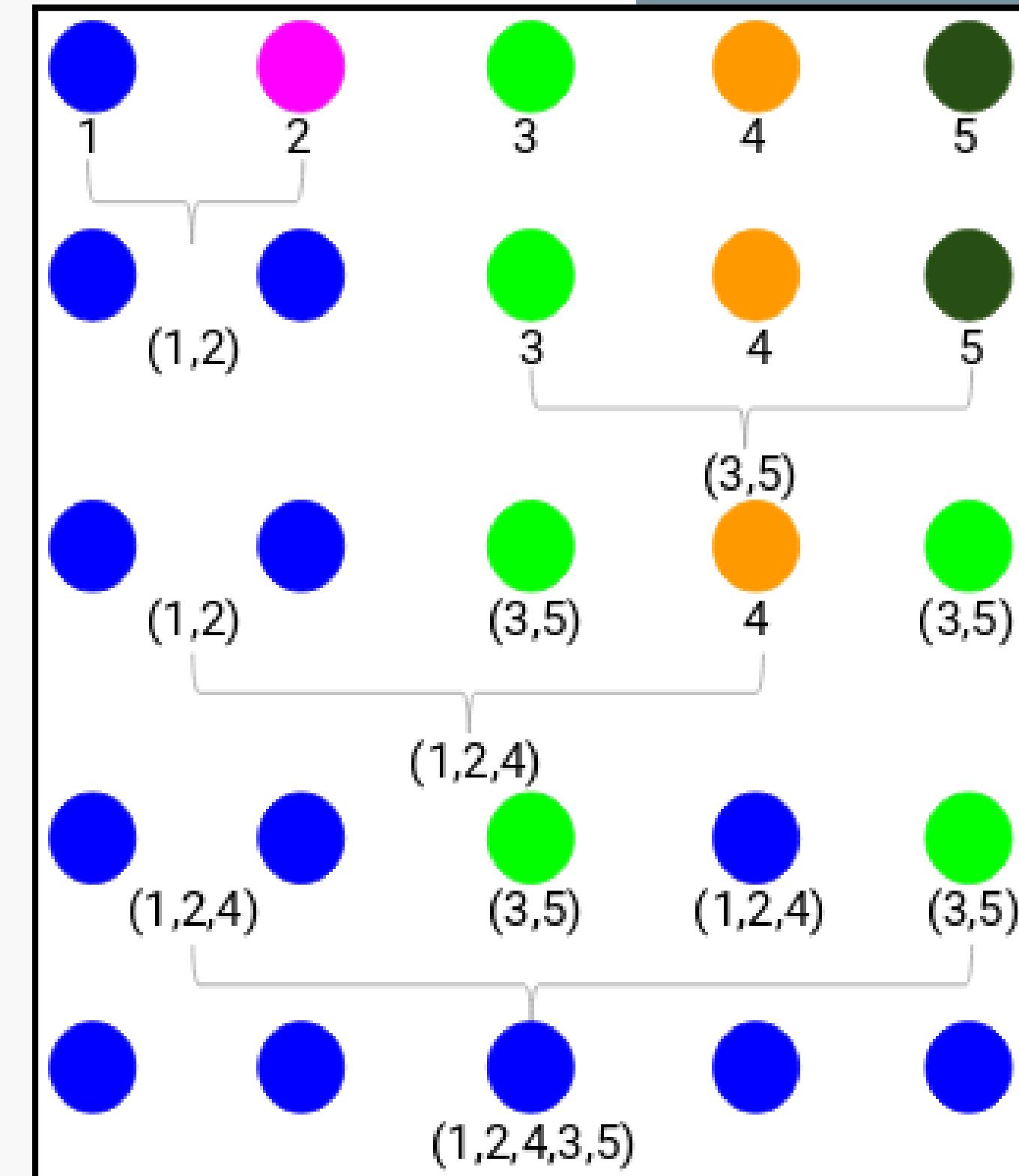
- Trục tung: Độ cao của mỗi nhánh trên trục y biểu thị mức độ khác biệt giữa các cụm được tính toán thông qua thước đo sự khác biệt.
- Trục hoành: thứ tự index của các quan sát trong tập dữ liệu gốc.



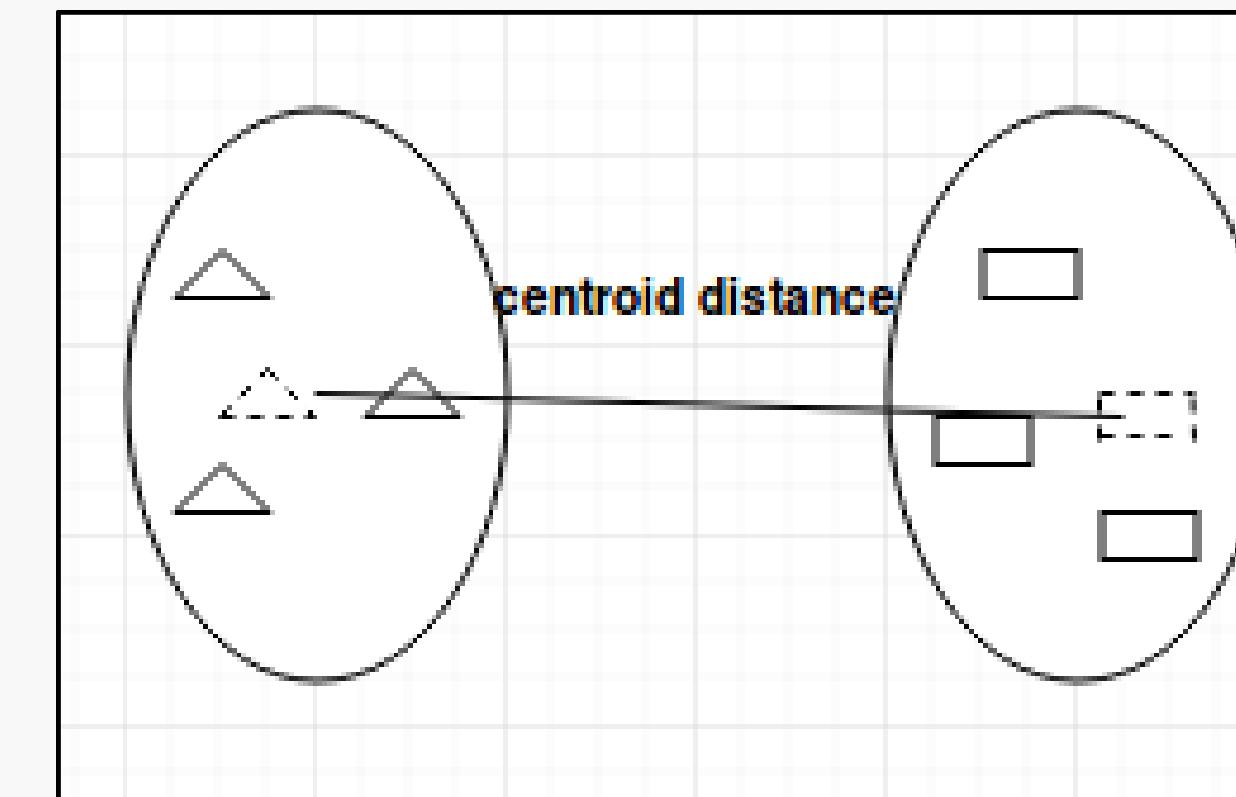
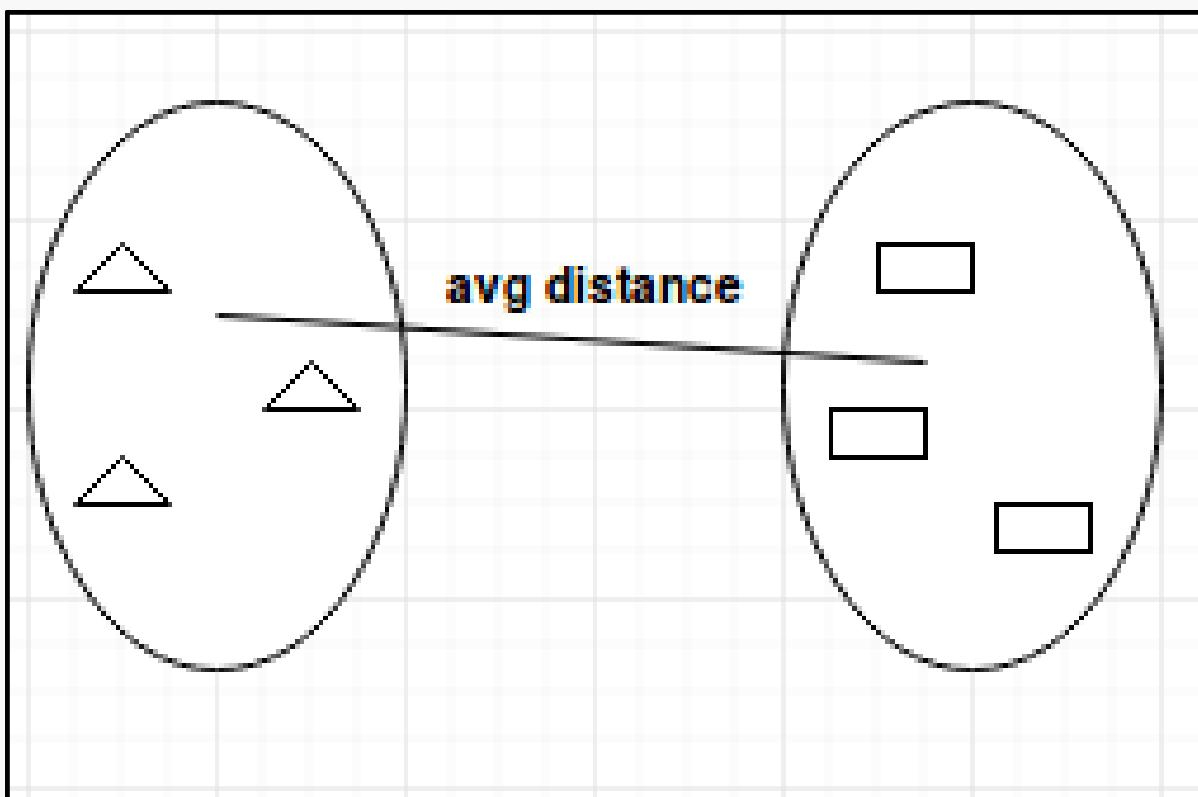
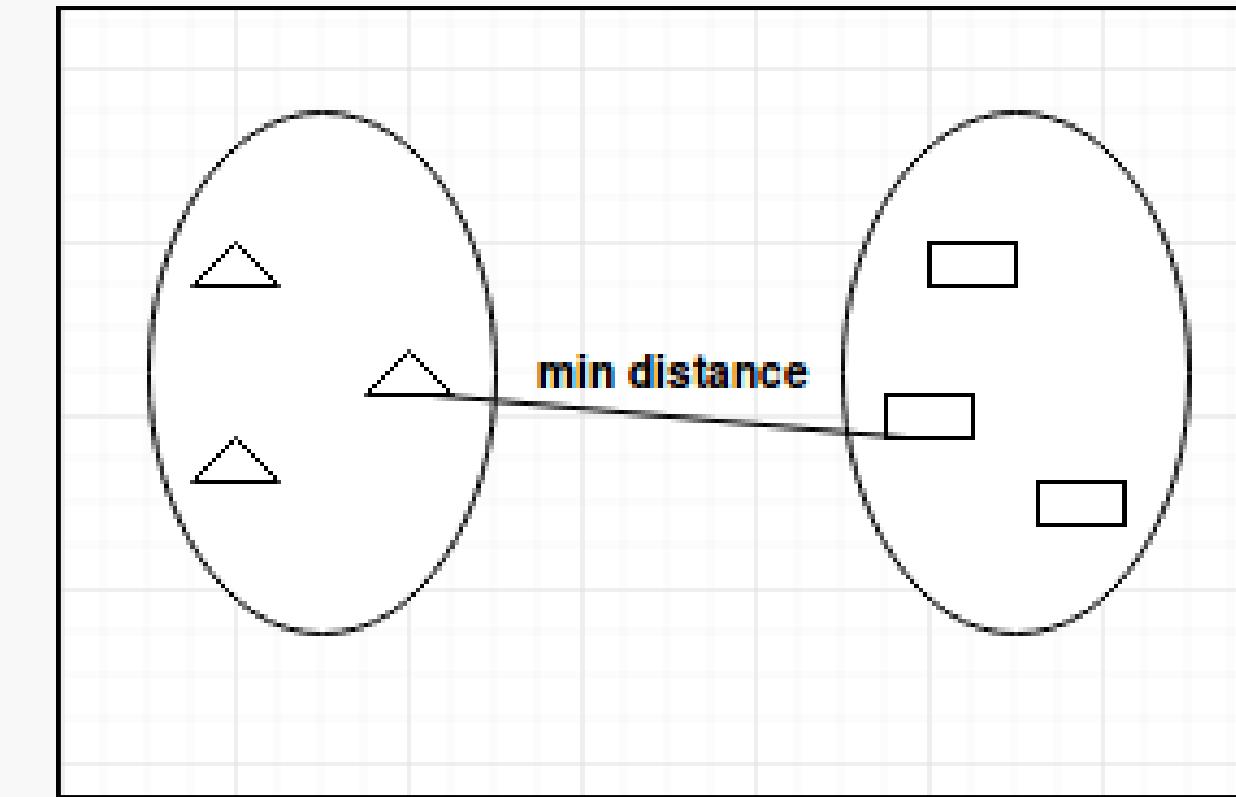
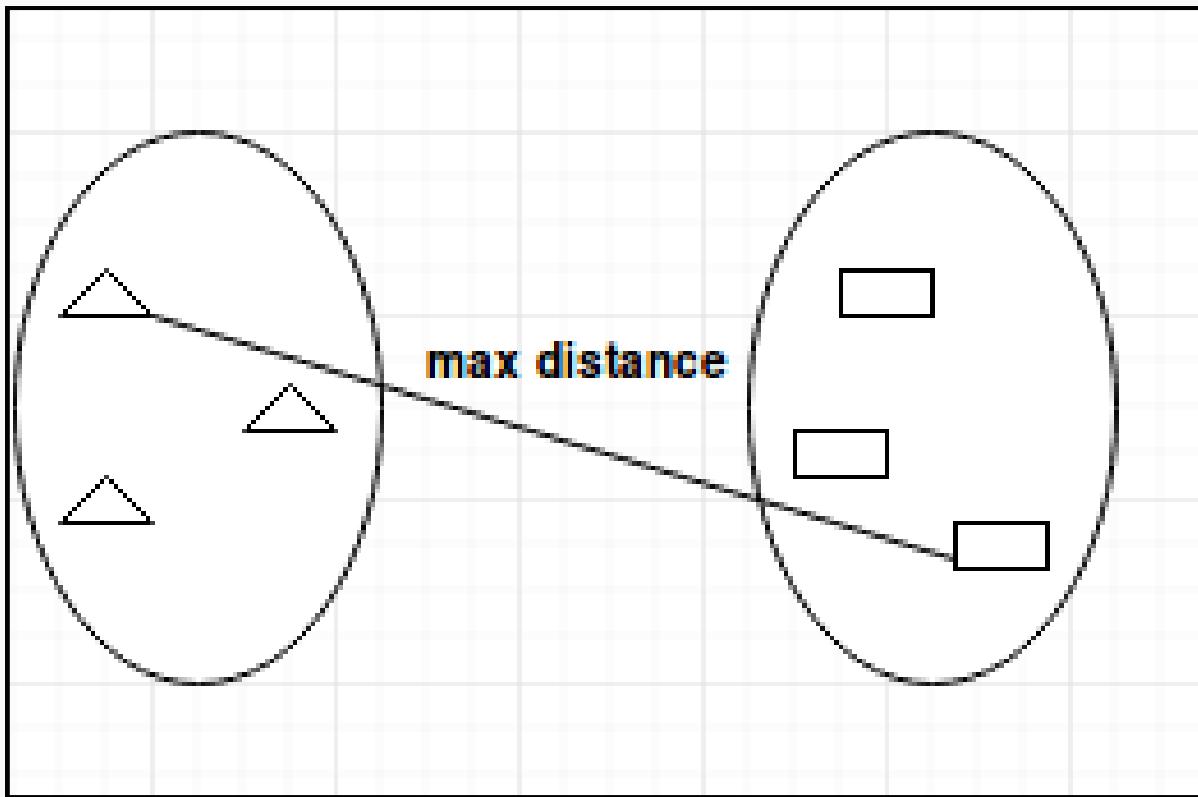
Chiến lược hợp nhất (agglomerative)

Hoạt động:

- Sẽ bắt đầu biểu diễn mỗi quan sát là một cụm đơn lẻ.
- N quan sát, thuật toán cần thực hiện $N-1$ bước để hợp nhất các cụm.
- Thuật toán gộp dần thành các cụm theo chiều từ dưới lên trên.
- Khoảng cách giữa hai cụm được đo lường thông qua một thước đo khoảng cách.



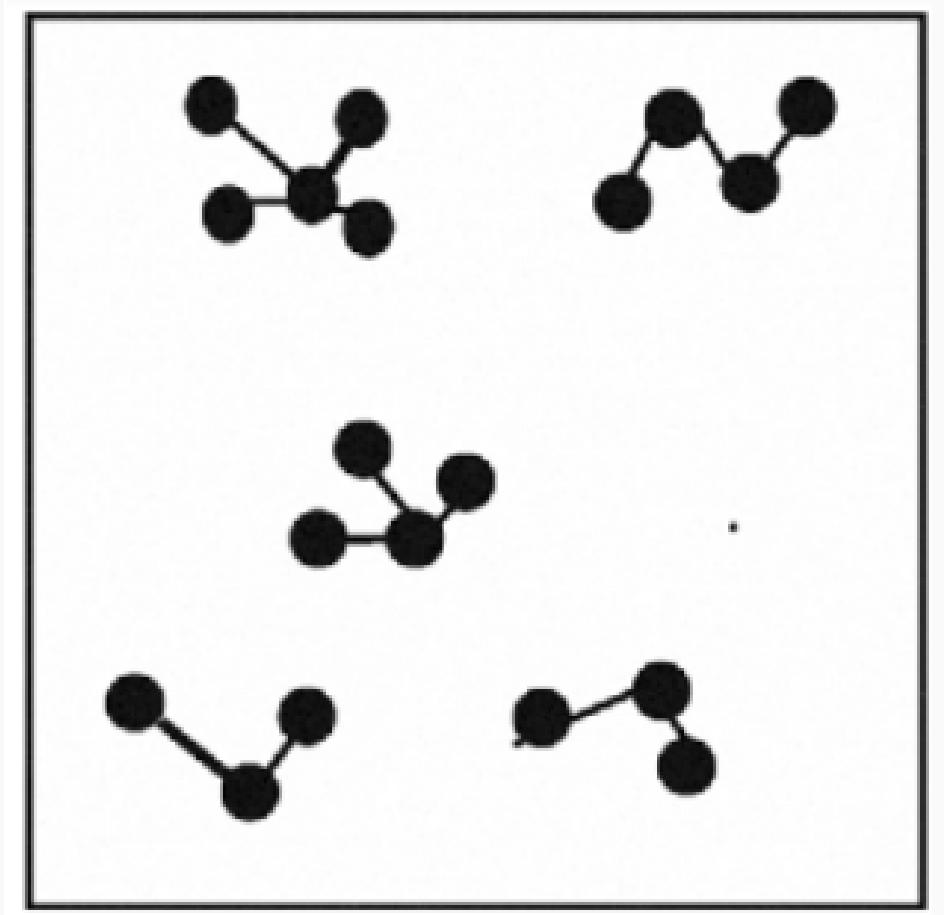
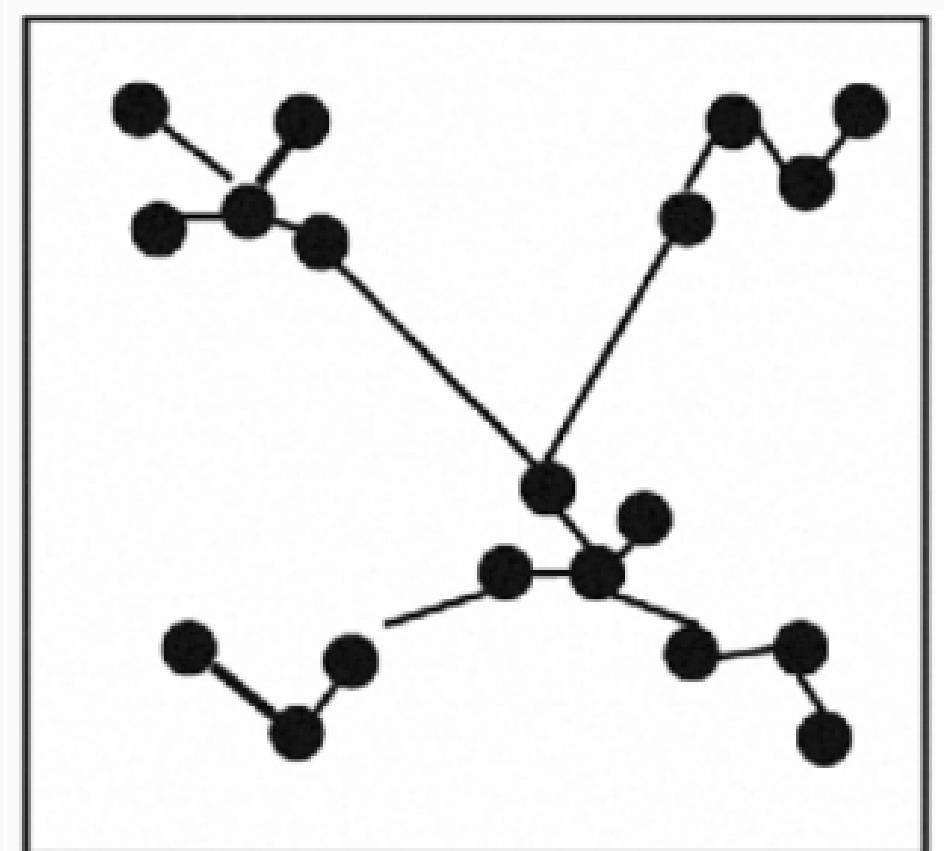
Đo khoảng cách



Chiến lược phân chia (divisive)

Hoạt động:

- Bắt đầu với một cụm duy nhất chứa tất cả các điểm dữ liệu.
- Tìm điểm chia tốt nhất trong cụm hiện tại, ví dụ như điểm chia tối đa hóa khoảng cách giữa các cụm con.
- Chia tách cụm hiện tại thành hai cụm con dựa trên điểm chia đã tìm được.
- Lặp lại bước chia tách cho mỗi cụm con cho đến khi đạt đến số lượng cụm mong muốn hoặc không thể chia tách thêm được nữa.



Xây dựng mô hình

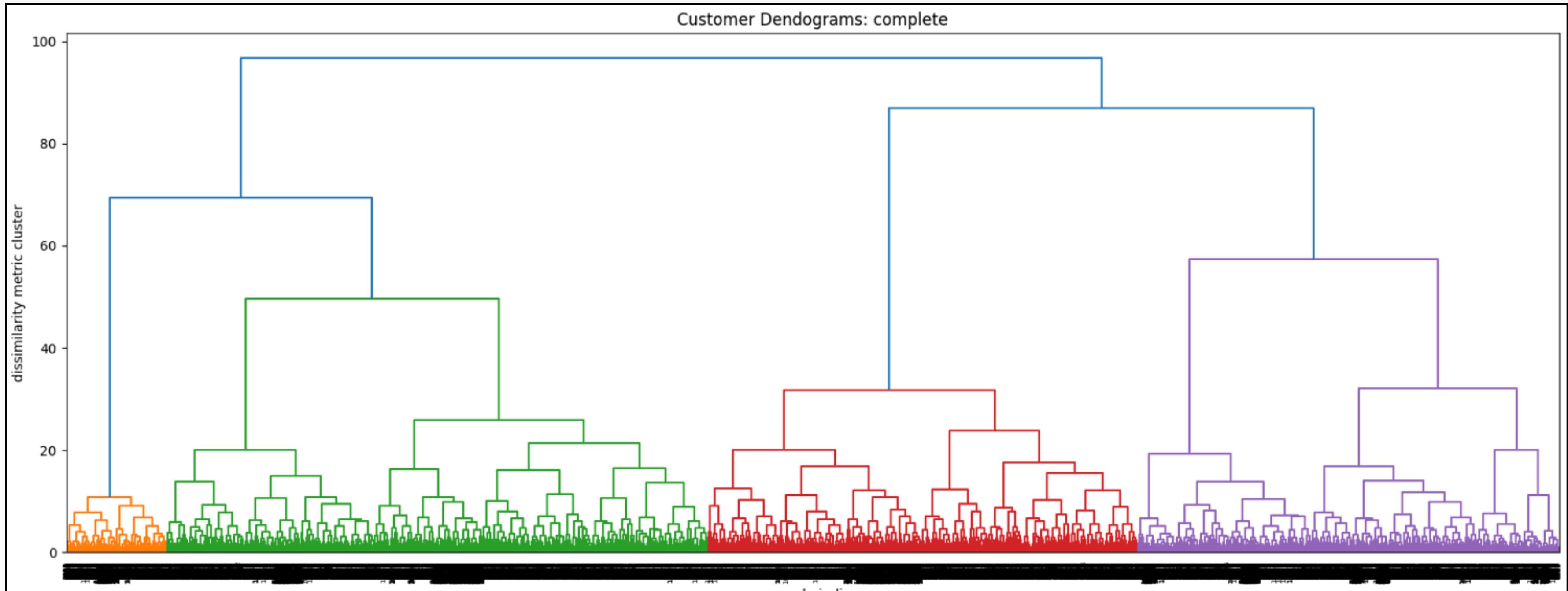
Vẽ biểu đồ Dendrogram

- Sử dụng chiến lược hợp nhất
- Đo khoảng cách giữa các cụm:
 - Ward linkage
 - Single linkage
 - Complete linkage
 - Group average

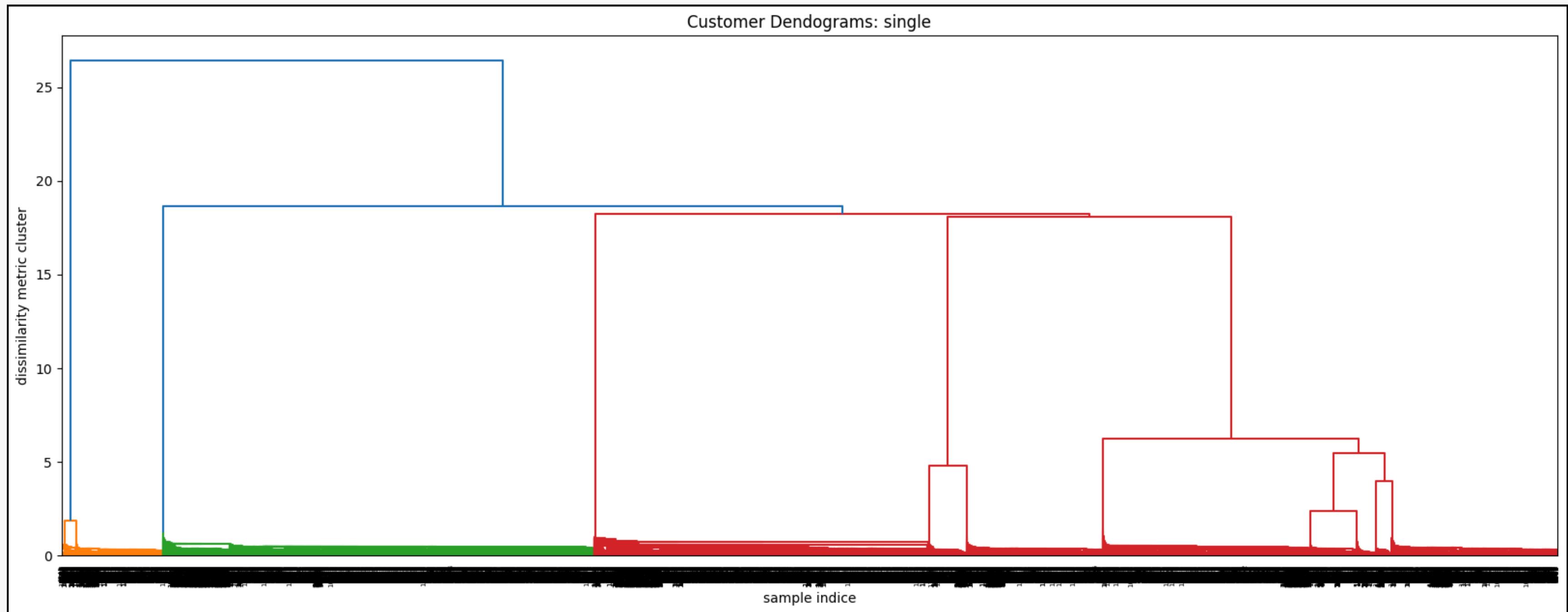
```
plt.figure(figsize=(20, 7))
plt.title("Customer Dendograms: average")
dend = shc.dendrogram(shc.linkage(data_players_tsne, method='average'))
plt.axhline(40, linestyle='--')
plt.xlabel('sample indice')
plt.ylabel('dissimilarity metric cluster')
✓ 4m 16.5s
```

```
plt.figure(figsize=(20, 7))
plt.title("Customer Dendograms: Ward linkage")
dend = shc.dendrogram(shc.linkage(players, method='ward'))
plt.axhline(12, linestyle='--')
plt.xlabel('sample indice')
plt.ylabel('dissimilarity metric cluster')
```

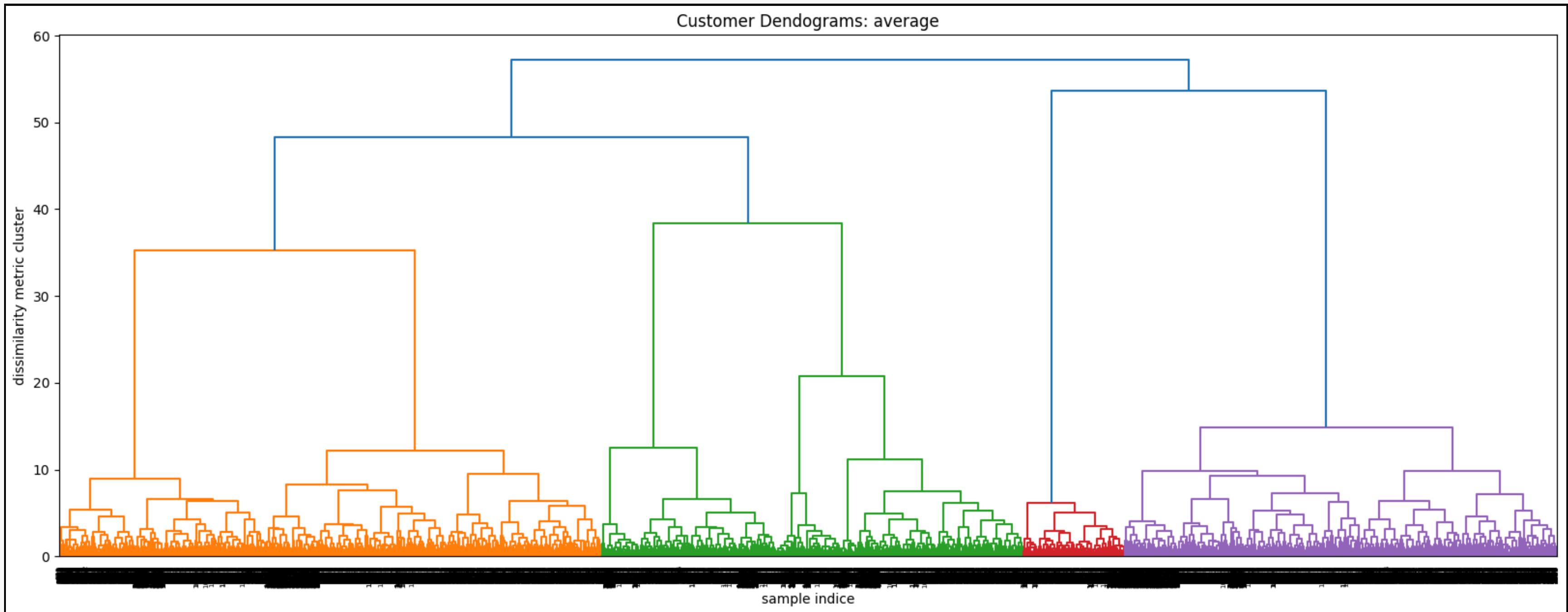
Xây dựng mô hình



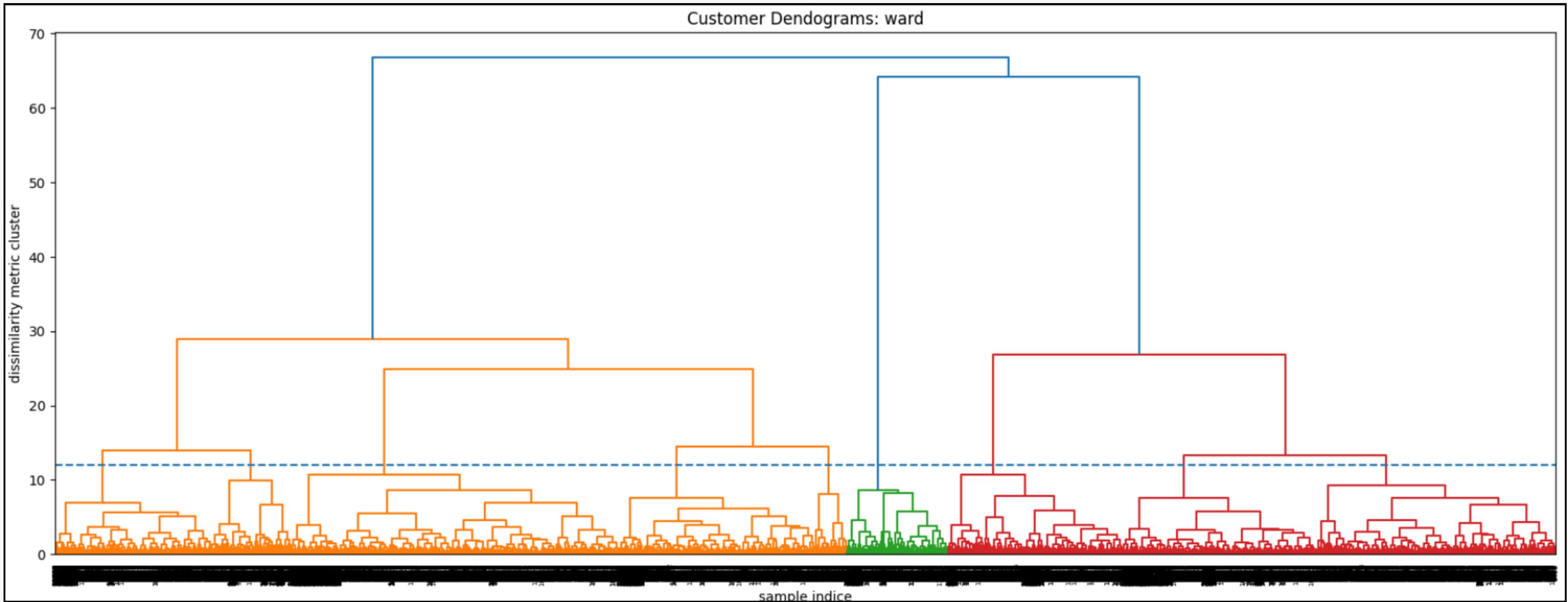
Xây dựng mô hình



Xây dựng mô hình



Xây dựng mô hình



Xây dựng mô hình

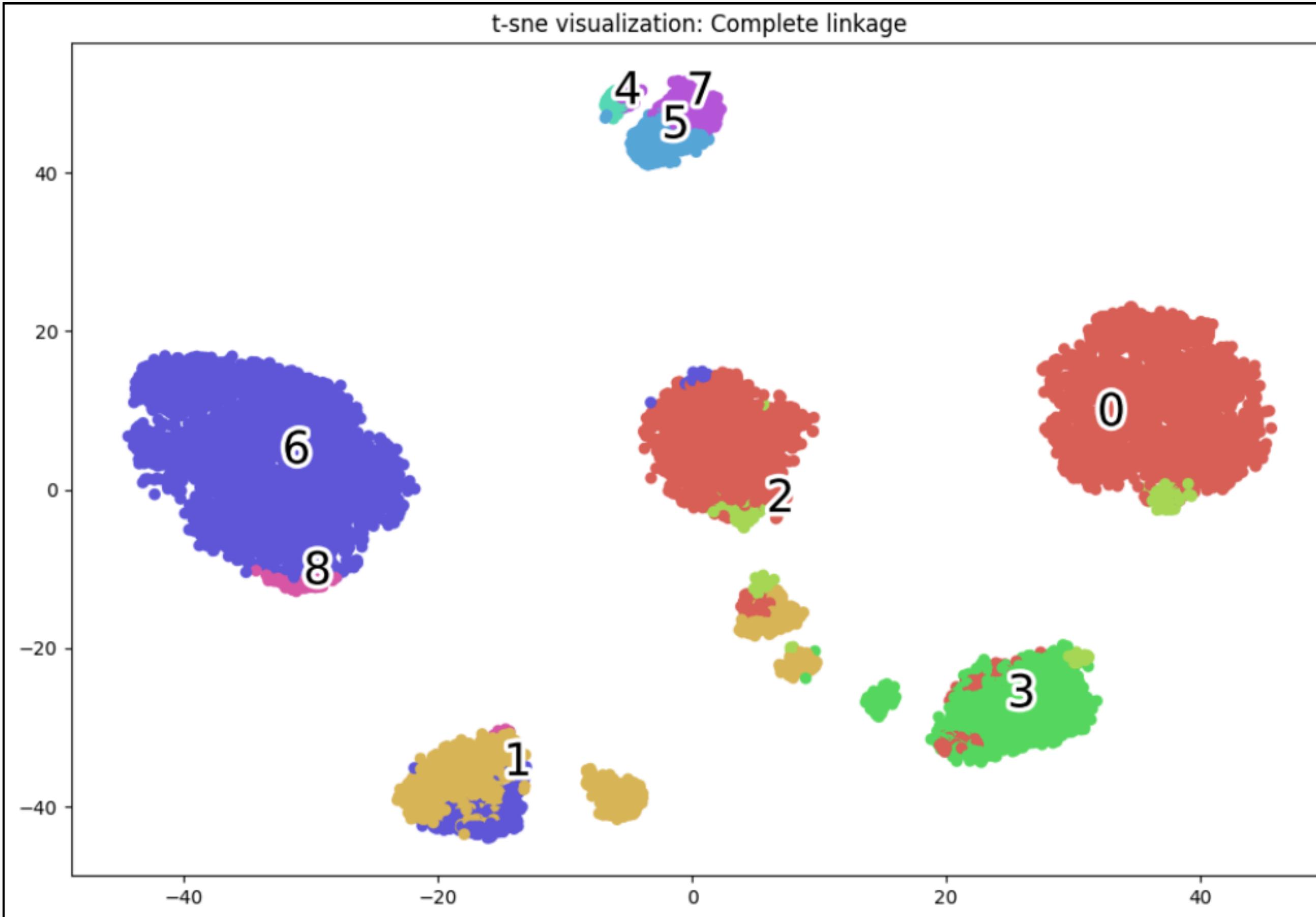
Gom nhóm và trực quan

- Gom nhóm các cụm theo số cụm mong muốn.
- Trực quan các cụm trên dữ liệu đã được giảm chiều.

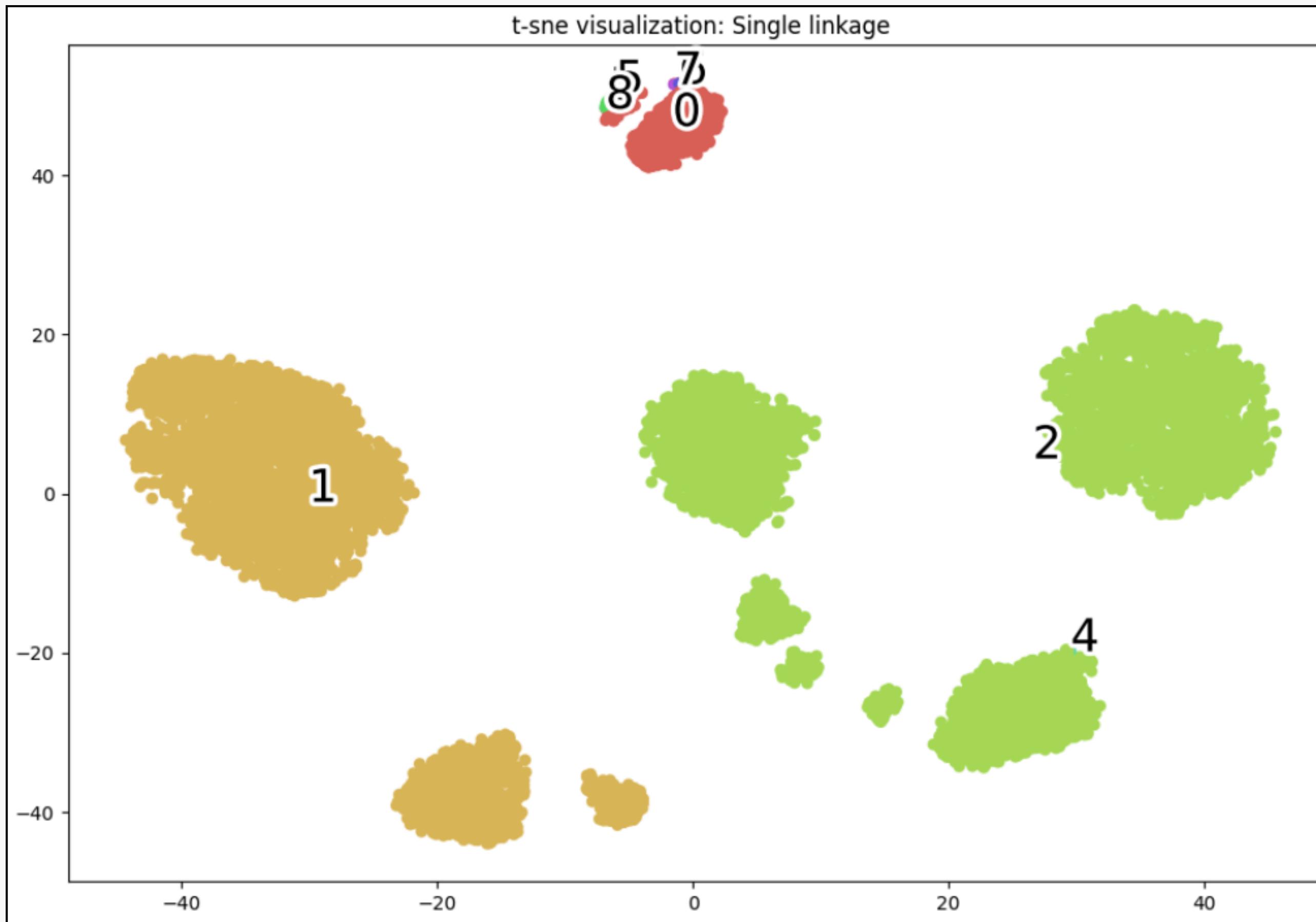
```
cluster = AgglomerativeClustering(n_clusters=9, linkage='ward')
labels_ward = cluster.fit_predict(players)
```

```
def _plot_kmean_scatter(X, labels, title):
    num_classes = len(np.unique(labels))
    palette = np.array(sns.color_palette("hls", num_classes))
    fig = plt.figure(figsize=(12, 8))
    ax = plt.subplot()
    sc = ax.scatter(X[:,0], X[:,1], lw=0, s=40, c=palette[labels.astype(int)])
    txts = []
    for i in range(num_classes):
        xtext, ytext = np.median(X[labels == i, :], axis=0)
        txt = ax.text(xtext, ytext, str(i), fontsize=24)
        txt.set_path_effects([
            PathEffects.Stroke(linewidth=5, foreground="w"),
            PathEffects.Normal()])
        txts.append(txt)
    plt.title('t-sne visualization: '+title)
```

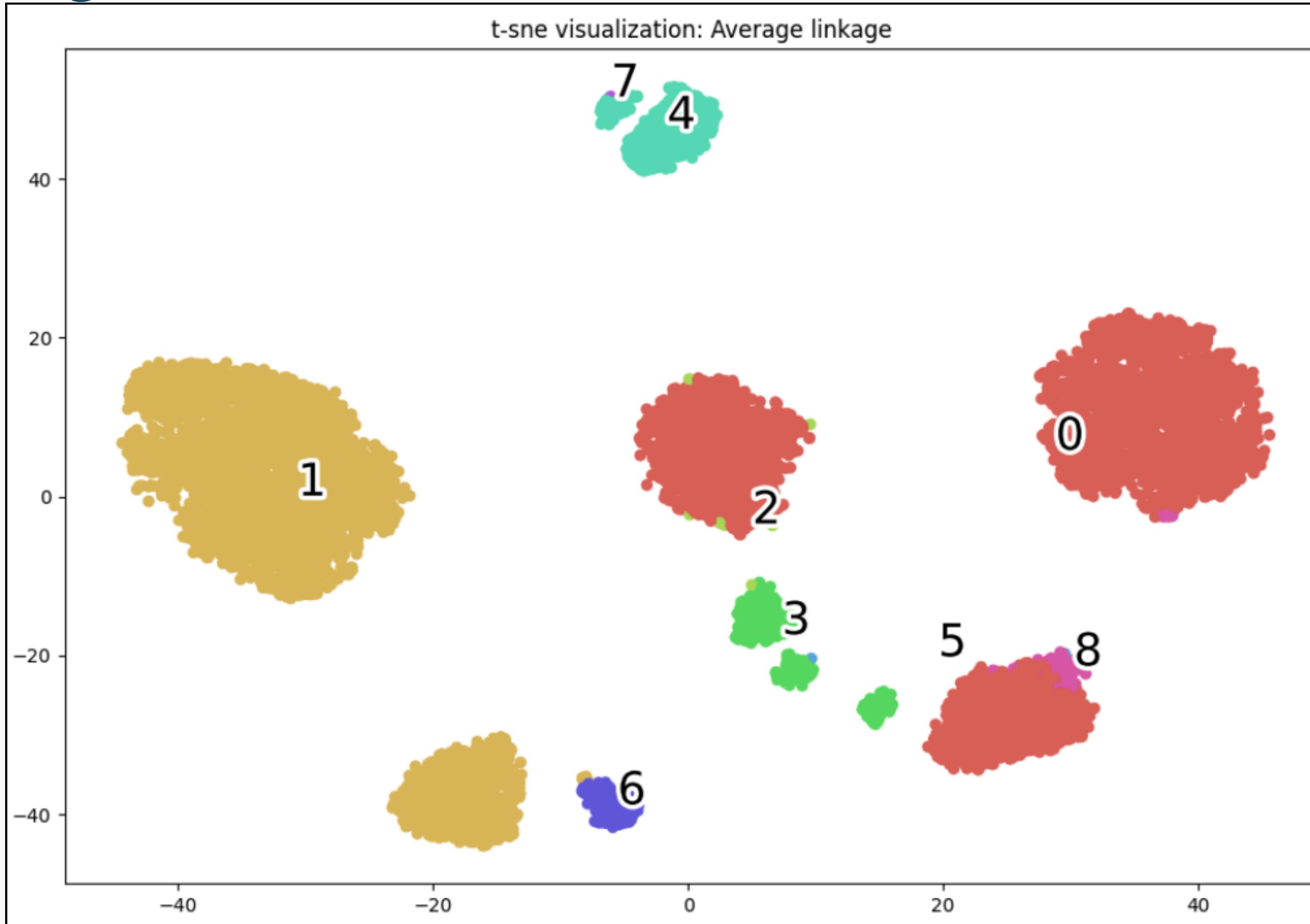
Xây dựng mô hình



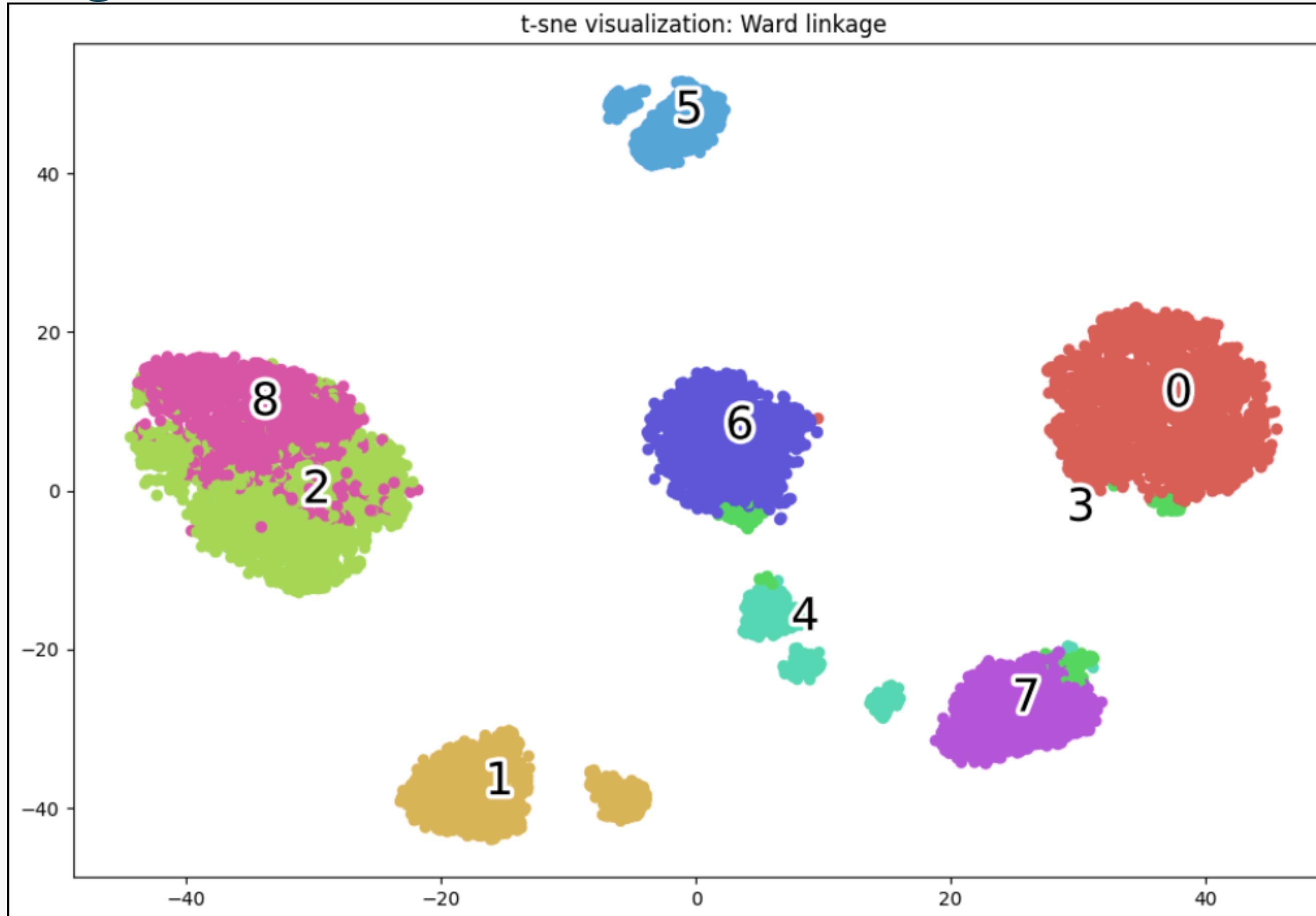
Xây dựng mô hình



Xây dựng mô hình



Xây dựng mô hình



Xây dựng mô hình

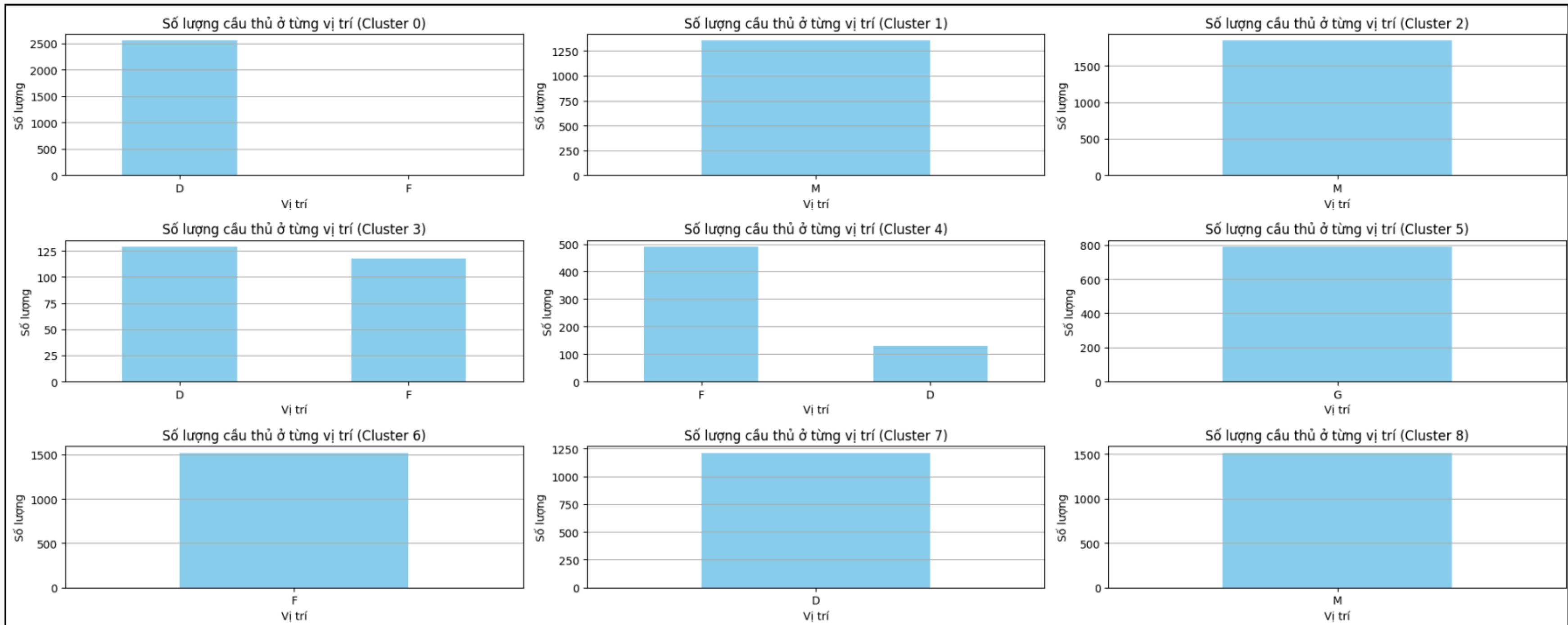
Khảo sát tính chất cụm

- Giá trị thị trường.
- Chỉ số thi đấu.
- Tuổi, chiều cao,

```
plt.figure(figsize=(20, 8))
for i, cluster_data in enumerate(clusters):
    positions = cluster_data['position']
    position_counts = positions.value_counts()
    plt.subplot(3, 3, i+1)
    position_counts.plot(kind='bar', color='skyblue')
    plt.title(f'Số lượng cầu thủ ở từng vị trí (Cluster {i})')
    plt.xlabel('Vị trí')
    plt.ylabel('Số lượng')
    plt.xticks(rotation=0)
    plt.grid(axis='y')
```

```
plt.subplot(1, 5, 3)
boxplot_scores = plt.boxplot(clusters_scores, labels=['Cluster 0', 'Cluster 1', 'Cluster 2',
|   'Cluster 3', 'Cluster 4', 'Cluster 5', 'Cluster 6', 'Cluster 7', 'Cluster 8'], patch_artist=True)
for patch, color in zip(boxplot_scores['boxes'], box_colors):
    patch.set_facecolor(color)
plt.title('Điểm số trung bình của cầu thủ trong các cluster')
plt.xlabel('Cluster')
plt.ylabel('Điểm số trung bình')
plt.xticks(rotation=90)
plt.xticks(rotation=90)
plt.grid(True)
```

Xây dựng mô hình



Nhận xét:

- Việc phân cụm có ảnh hưởng bởi vị trí đá của cầu thủ



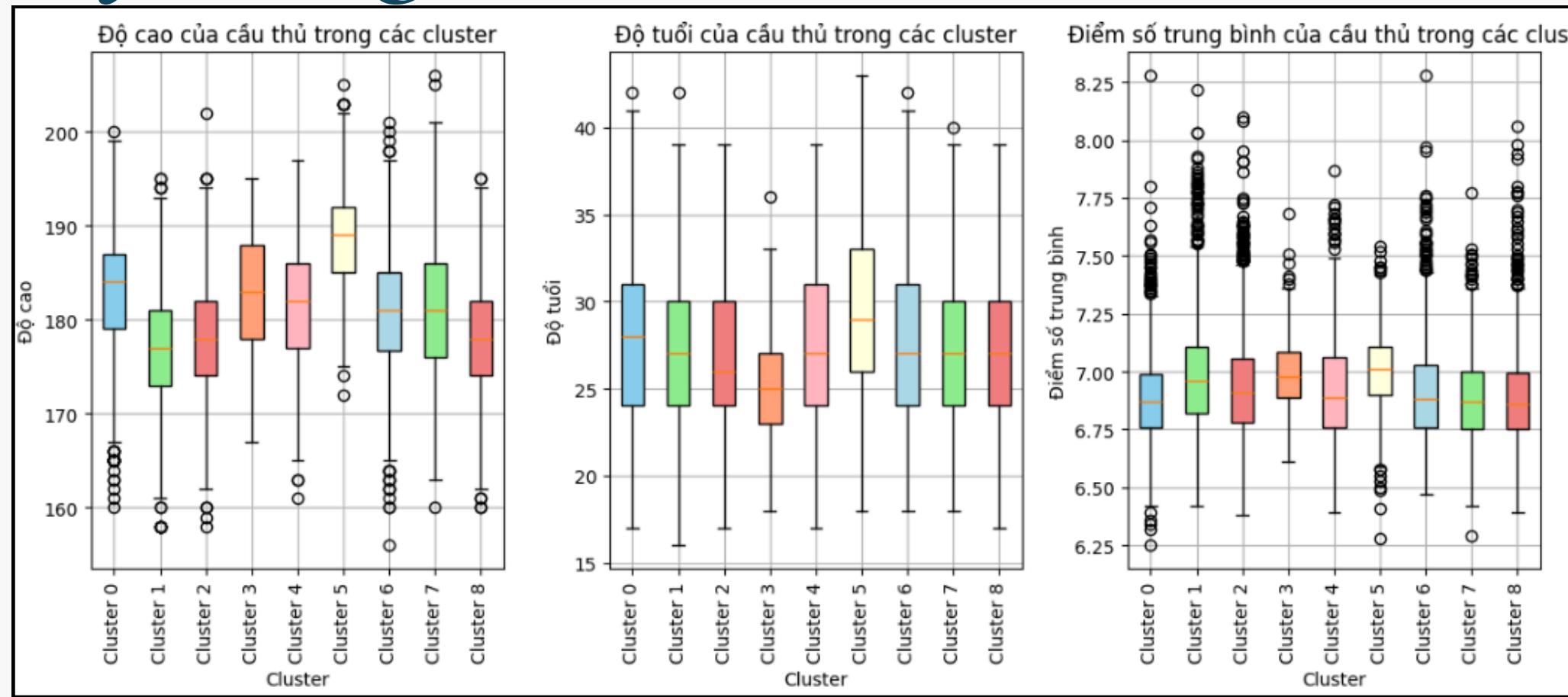
Hậu vệ: 0, 3, 4, 7

Tiền đạo: 3, 4, 6

Tiền vệ: 1, 2, 8

Thủ môn: 5

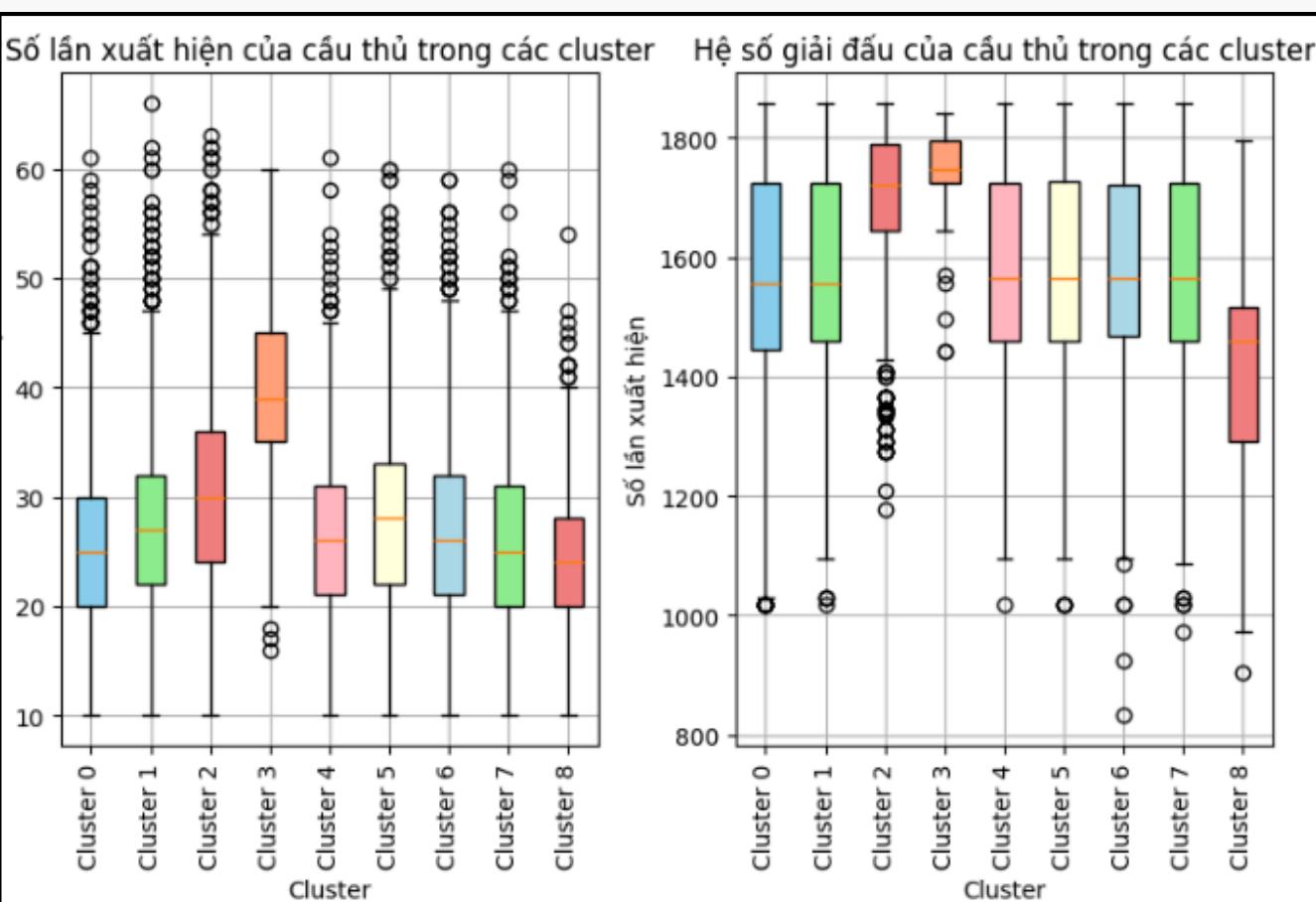
Xây dựng mô hình



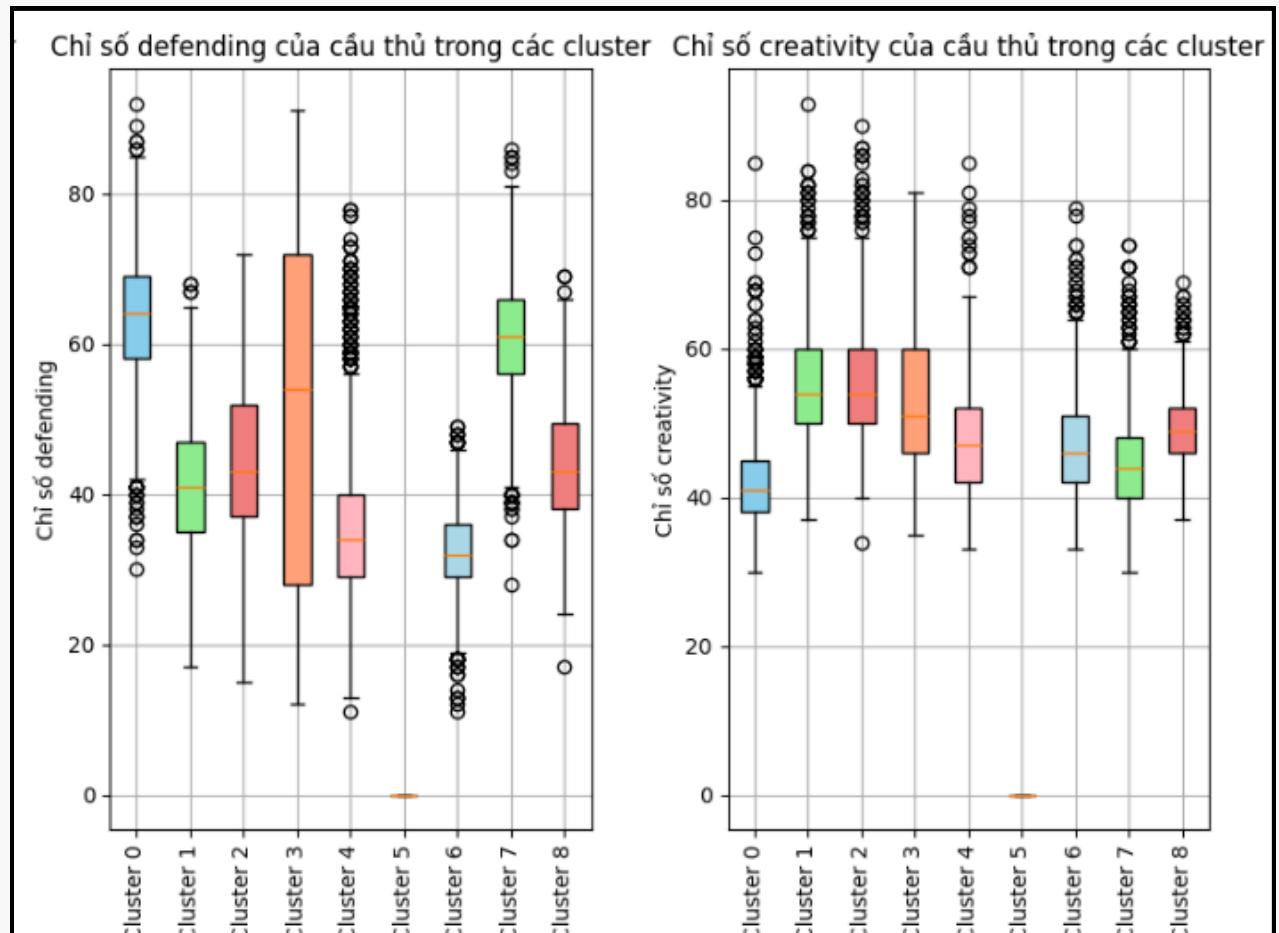
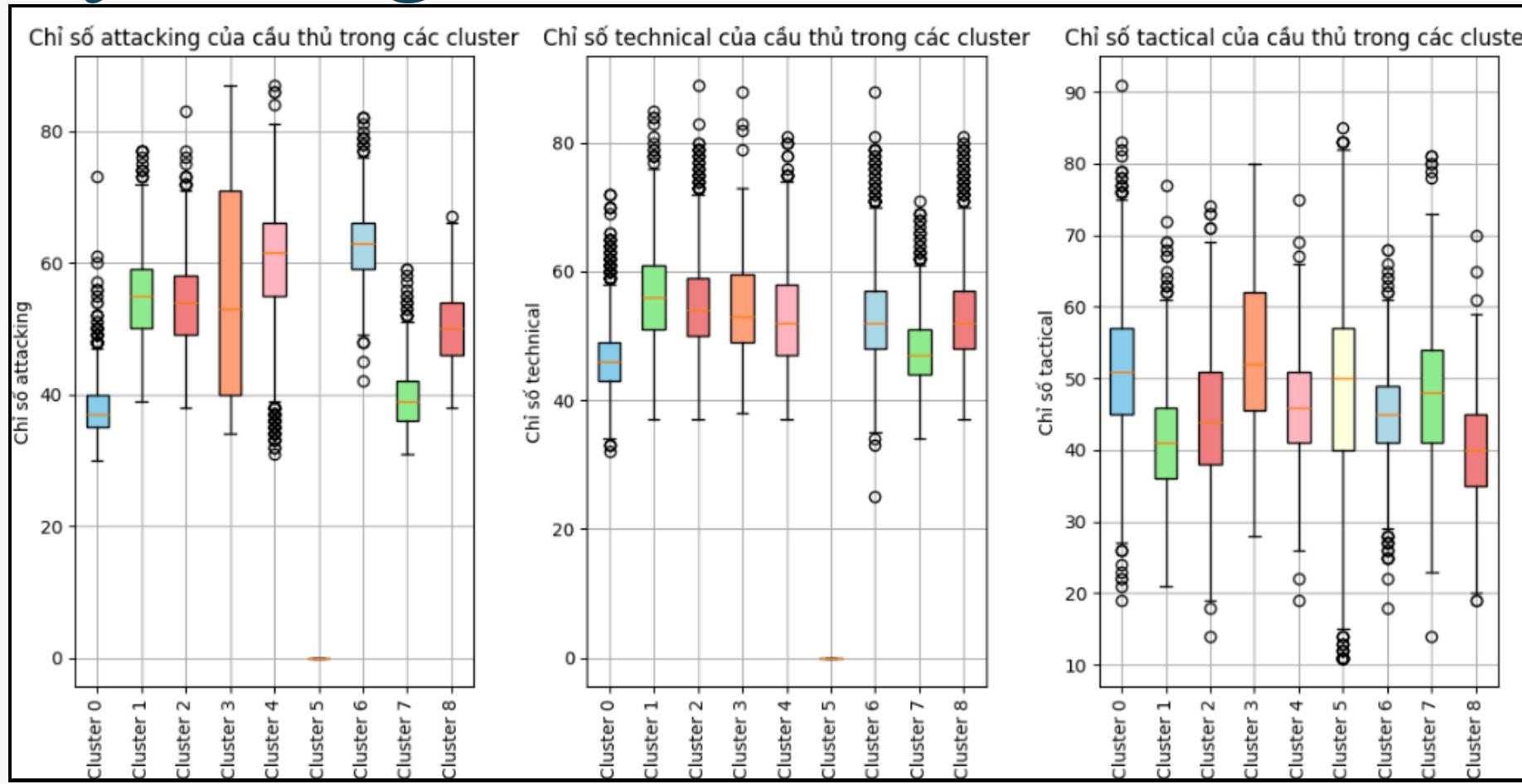
Hậu vệ: 0, 3, 4, 7
Tiền đạo: 3, 4, 6
Tiền vệ: 1, 2, 8
Thủ môn: 5

Nhận xét:

- Các cụm Hậu vệ, Tiền đạo: khá tương đồng. Ở cụm 3 thì Hậu vệ và Tiền đạo sẽ nhỉnh hơn về độ tuổi, tần suất xuất hiện và hệ số giải đấu.
- Cụm về Thủ môn: độ cao, tuổi trung bình lớn.
- Cụm về Tiền vệ: ở cụm 1, 2 thì giải đấu và số lần xuất hiện cao hơn.



Xây dựng mô hình



Hậu vệ: 0, 3, 4, 7
Tiền đạo: 3, 4, 6
Tiền vệ: 1, 2, 8
Thủ môn: 5

Nhận xét:

Các cụm về Hậu vệ(D) :

- Chỉ số Defending, Tactical cao. Chỉ số về Attacking, Technical, Creativity thấp.

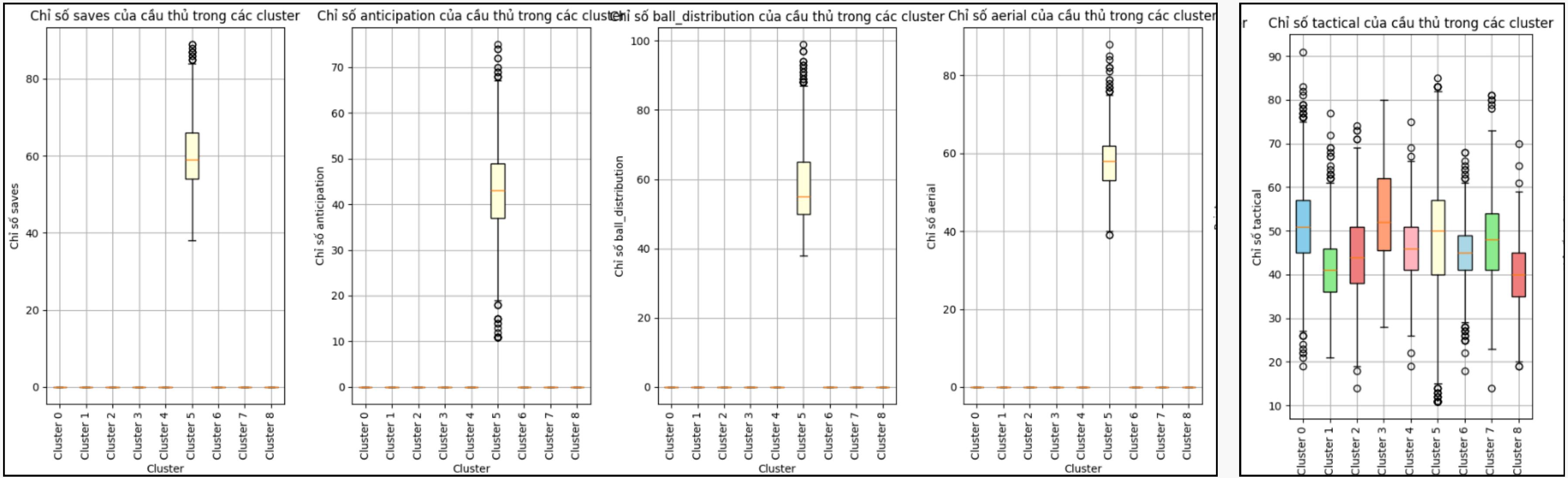
Cụm về Tiền đạo(F):

- các chỉ số đều mức thấp ở cụm 6.
- cụm 3 cũng mạnh về chỉ số Tactical và Defending từ đó thấy được sự toàn diện về chỉ số của cụm 3.

Các cụm về Tiền vệ(M):

- cụm 1, 2 có chỉ số Attacking, Technical và Creativity khá sao.
- Tiền vệ trong cụm 8 có chỉ số thấp.

Xây dựng mô hình

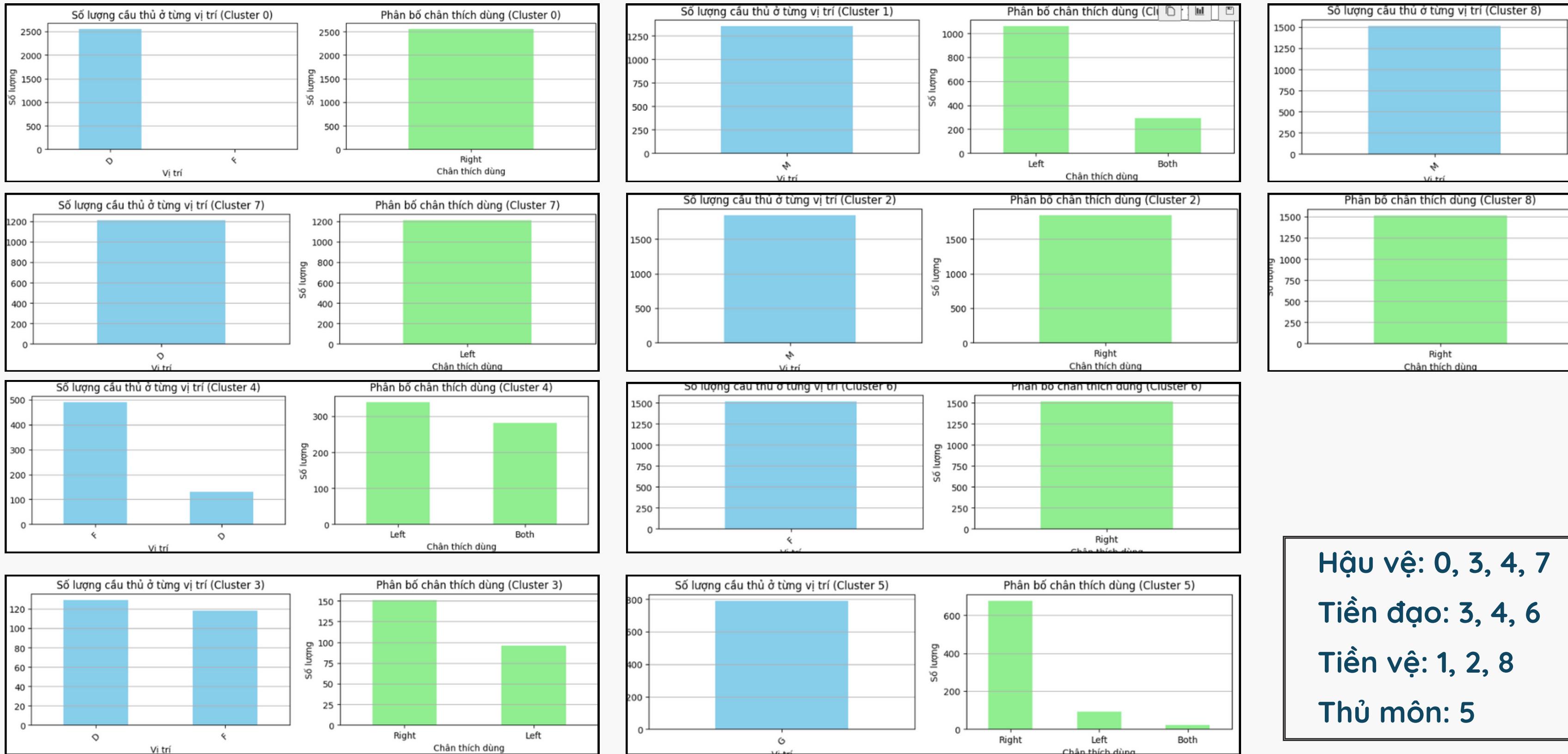


Hậu vệ: 0, 3, 4, 7
Tiền đạo: 3, 4, 6
Tiền vệ: 1, 2, 8
Thủ môn: 5

Nhận xét:

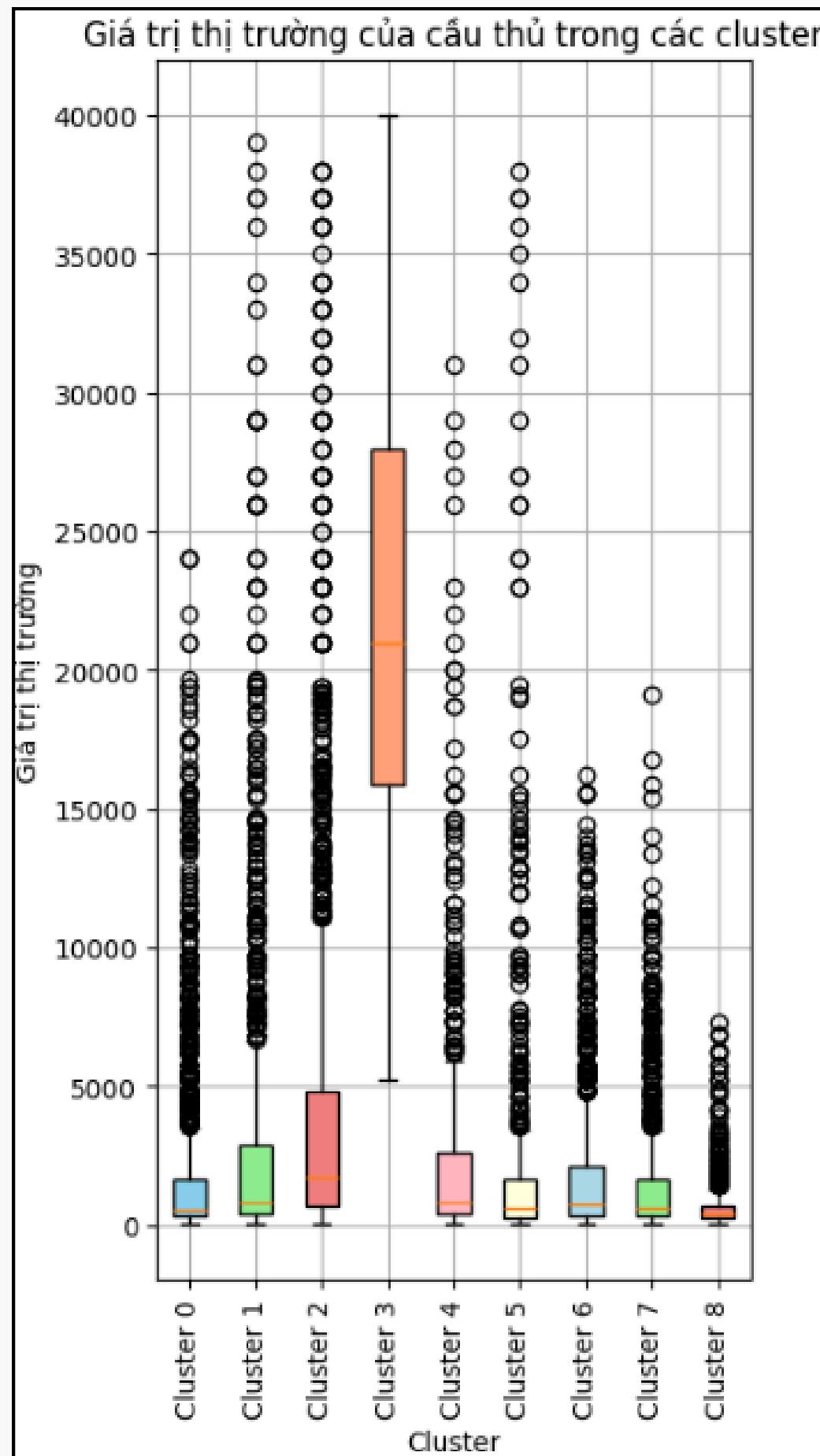
- Các giá trị chỉ số của các thủ môn ít tách biệt do đó chỉ có duy nhất 1 cụm Thủ môn.

Xây dựng mô hình



Hậu vệ: 0, 3, 4, 7
Tiền đạo: 3, 4, 6
Tiền vệ: 1, 2, 8
Thủ môn: 5

Xây dựng mô hình



Hậu vệ: 0, 3, 4, 7
Tiền đạo: 3, 4, 6
Tiền vệ: 1, 2, 8
Thủ môn: 5

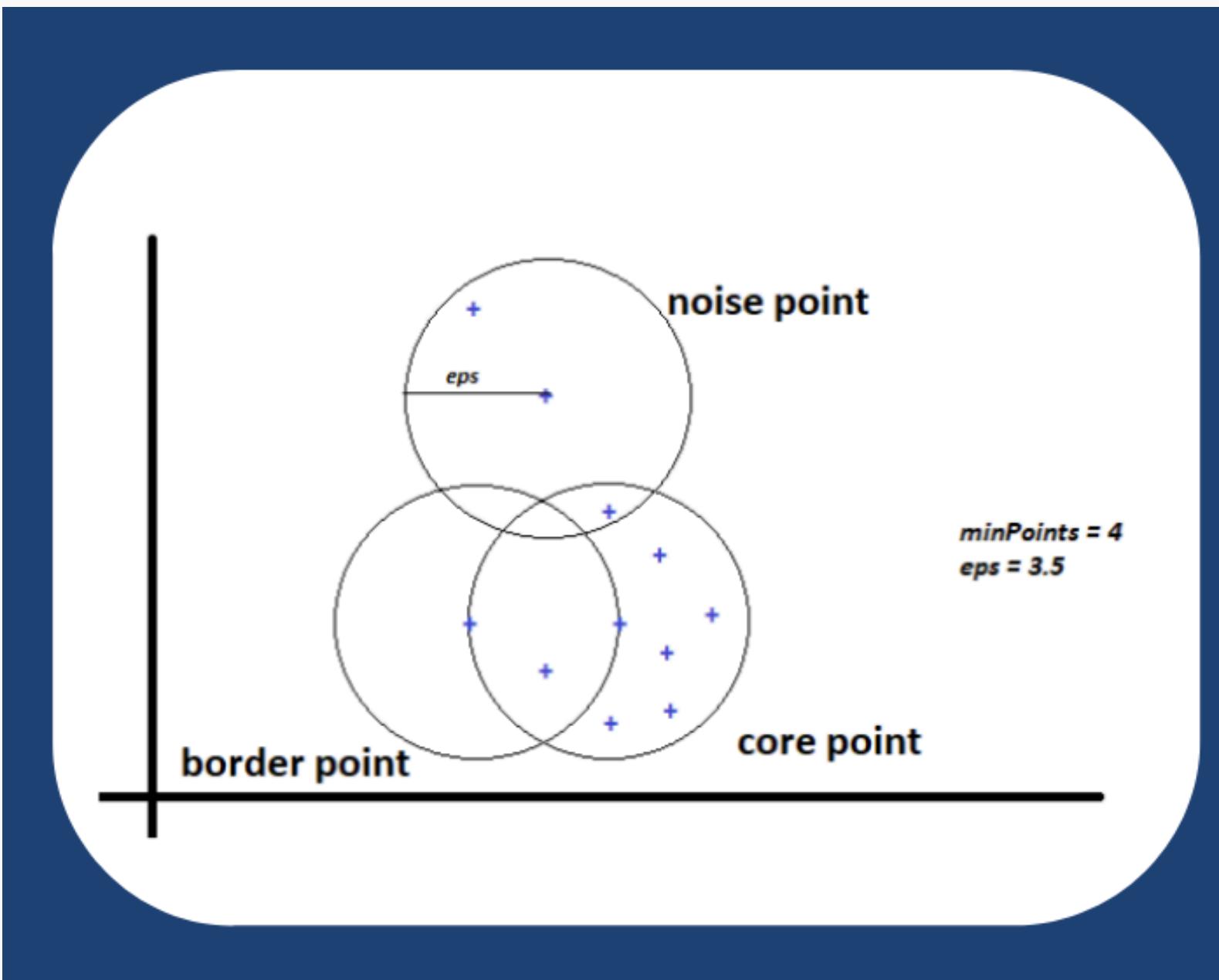
Nhận xét:

- Hậu vệ và Tiền đạo ở cụm 3 có giá trị thị trường rất cao và trung bình cao ở cụm 4
- Hậu vệ ở cụm 0, 7 có giá trị mức trung bình.
- Tiền đạo ở cụm 6 có giá trị trung bình cao.
- Thủ môn có giá trị ở mức thấp.

Kết luận

- Phương pháp phân cụm phân cấp Hierarchical Clustering cho ra kết quả các cụm có tính chất và phân bố khác nhau tùy theo phương pháp đo khoảng cách giữa 2 cụm và lựa chọn thước đo về sự khác biệt giữa các cụm
- Hierarchical Clustering cho ra kết quả phân cụm thể hiện tốt mối quan hệ giữa các cụm.
- Các cụm thu được:
 - Cụm 0: Là Hậu vệ, defending|tactical khá cao, attacking|technical|creativity thấp, chân phải, giá trị trung bình
 - Cụm 1: Là Tiền vệ, hay ra sân, giải đấu lớn, attacking|technical|creativity khá cao, defending|tactical trung bình, chân trái|số ít thuận 2 chân, giá trị trung bình cao
 - Cụm 2: Là Tiền vệ, hay ra sân, giải đấu lớn, attacking|technical|creativity trung bình, defending|tactical khá cao, chân phải, giá trị trung bình cao
 - Cụm 3: Là Tiền đạo/Hậu vệ, trẻ, hay ra sân, giải đấu lớn, chỉ số cao toàn diện, chân phải, giá trị cao
 - Cụm 4: Là Tiền đạo/Hậu vệ, attacking|technical|creativity cao, chân trái, giá trị trung bình cao
 - Cụm 5: Là Thủ môn, cao, lớn tuổi, tactical cao, chân phải|số ít chân trái hoặc cả 2 chân
 - Cụm 6: Là Tiền đạo, chỉ số thấp, chân phải, giá trị trung bình thấp
 - Cụm 7: Là Hậu vệ, defending|tactical trung bình, attacking|technical|creativity khá cao, chân trái, giá trị trung bình
 - Cụm 8: Là Tiền vệ, ít ra sân, giải đấu nhỏ, chỉ số thấp, chân phải, giá trị thấp

DBSCAN



Giới thiệu

- DBSCAN là một thuật toán để phân nhóm dựa trên mật độ. Nó có thể phát hiện ra các cụm có hình dạng và kích thước khác nhau từ một lượng lớn dữ liệu chứa nhiễu.

Các định nghĩa, tham số chính trong DBSCAN

- Vùng lân cận epsilon của một điểm dữ liệu P được định nghĩa là tập hợp tất cả các điểm dữ liệu nằm trong phạm vi bán kính epsilon.
- Epsilon: là giá trị khoảng cách được sử dụng để xác định vùng lân cận epsilon của bất kỳ điểm dữ liệu nào. Giá trị của epsilon được chọn bằng biểu đồ k-distance, epsilon đạt giá trị tốt nhất tại elbow point.

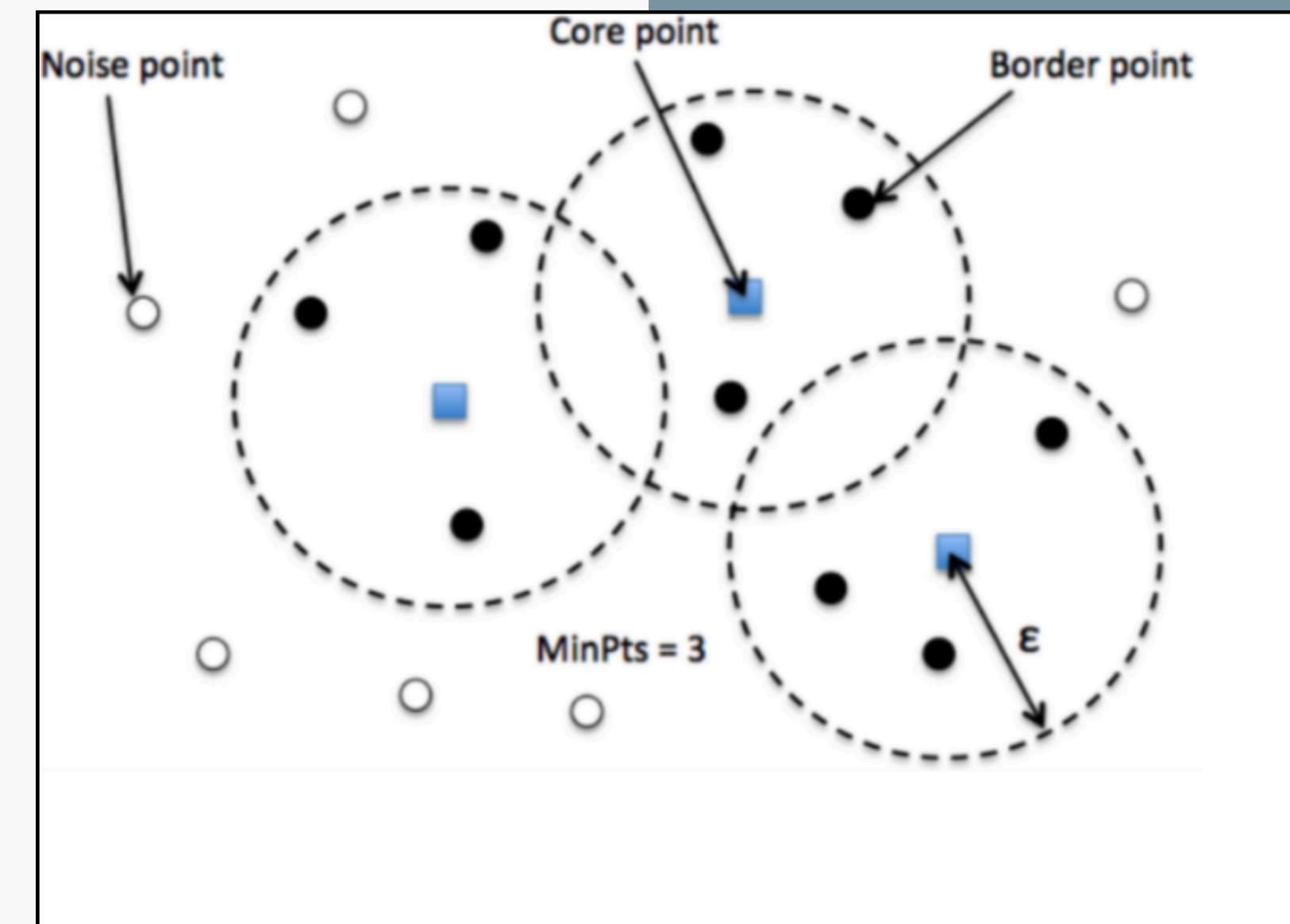
DBSCAN

Các định nghĩa, tham số chính trong DBSCAN

- MinPts: Là số điểm tối thiểu để nhóm lại với nhau nhằm xác định vùng lân cận epsilon. Số điểm min points không bao gồm điểm ở tâm. $\text{minPts} \geq D + 1$ (với D là số chiều của tập dữ liệu). Theo quy tắc chung thường chọn Min Points = $2 \times D$

Phân loại dạng điểm trong DBSCAN

- Core point: là một điểm có ít nhất minPts trong vùng lân cận epsilon của nó.
- Border point: là một điểm có ít nhất một core point trong vùng lân cận epsilon nhưng mật độ không đủ minPts điểm.
- Noise point: là điểm không phải core point hay border point.



DBSCAN

Các bước trong thuật toán

Bước 1:

- Lựa chọn 1 điểm bất kỳ. Sau đó xác định các core point và border point.



Bước 2:

- Chọn ngẫu nhiên core point (không nằm trong cụm nào cả) gán vào 1 cụm.
- Tìm các điểm lân cận của core point.
 - Nếu điểm đó là core point thì thêm vào cụm, tiếp tục mở rộng cụm từ core point này.
 - Nếu điểm đó là border point thì thêm vào cụm nhưng không mở rộng cụm từ điểm này



Bước 3:

- Lặp lại đệ quy toàn bộ quá trình để xác định một cụm mới.



DBSCAN

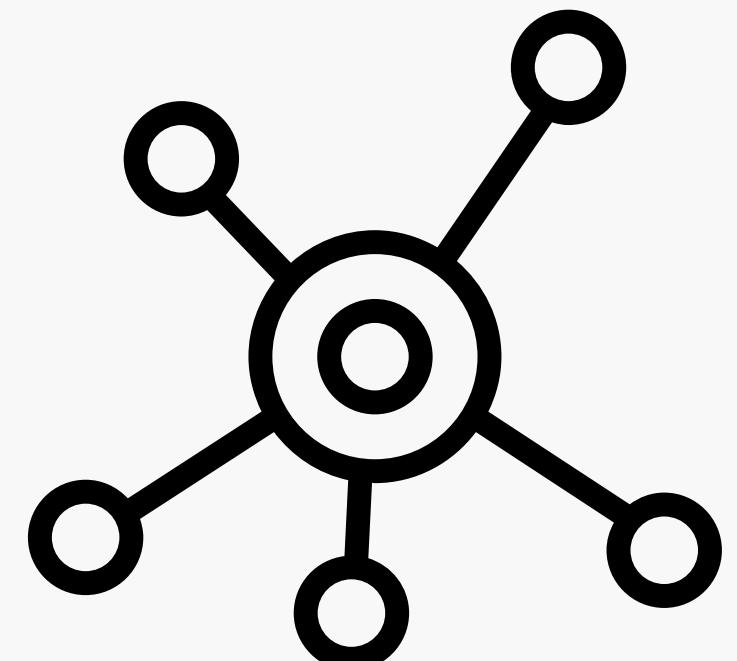


Ưu điểm

- Phân cụm theo mật độ của dữ liệu (bỏ qua các outlier).
- Tìm được các điểm không thuộc bất kỳ 1 cụm nào.
- Không cần biết trước số lượng cụm.

Nhược điểm

- Phải tìm được epsilon.
- Không phù hợp với bài toán mà kiểu dữ liệu thưa thớt hay tập dữ liệu phân bố không đồng đều



DBSCAN

Xây dựng mô hình

```
neighbors = 2*17
nbrs = NearestNeighbors(n_neighbors=neighbors).fit(players)
# Ma trận khoảng cách distances: (N, k)
distances, indices = nbrs.kneighbors(players)
# Lấy ra khoảng cách xa nhất từ phạm vi lảng giềng của mỗi điểm
distance_desc = sorted(distances[:, neighbors-1], reverse=True)
# Plot...
```

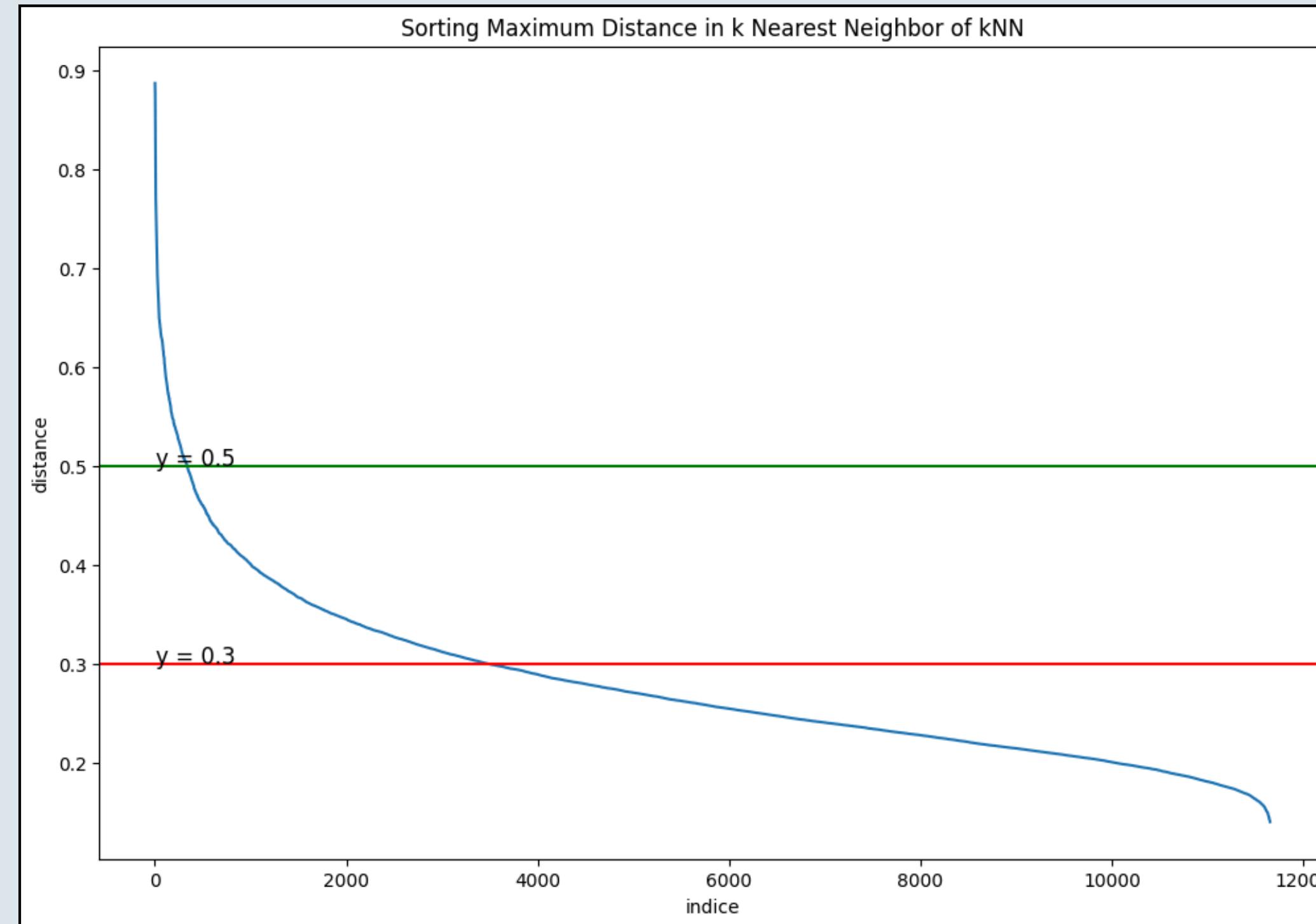
Code tìm epsilon

```
dbscan = DBSCAN(eps=0.4818, min_samples=2*17+1)
labels = dbscan.fit_predict(players)
# Plot
```

Code áp dụng DBSCAN

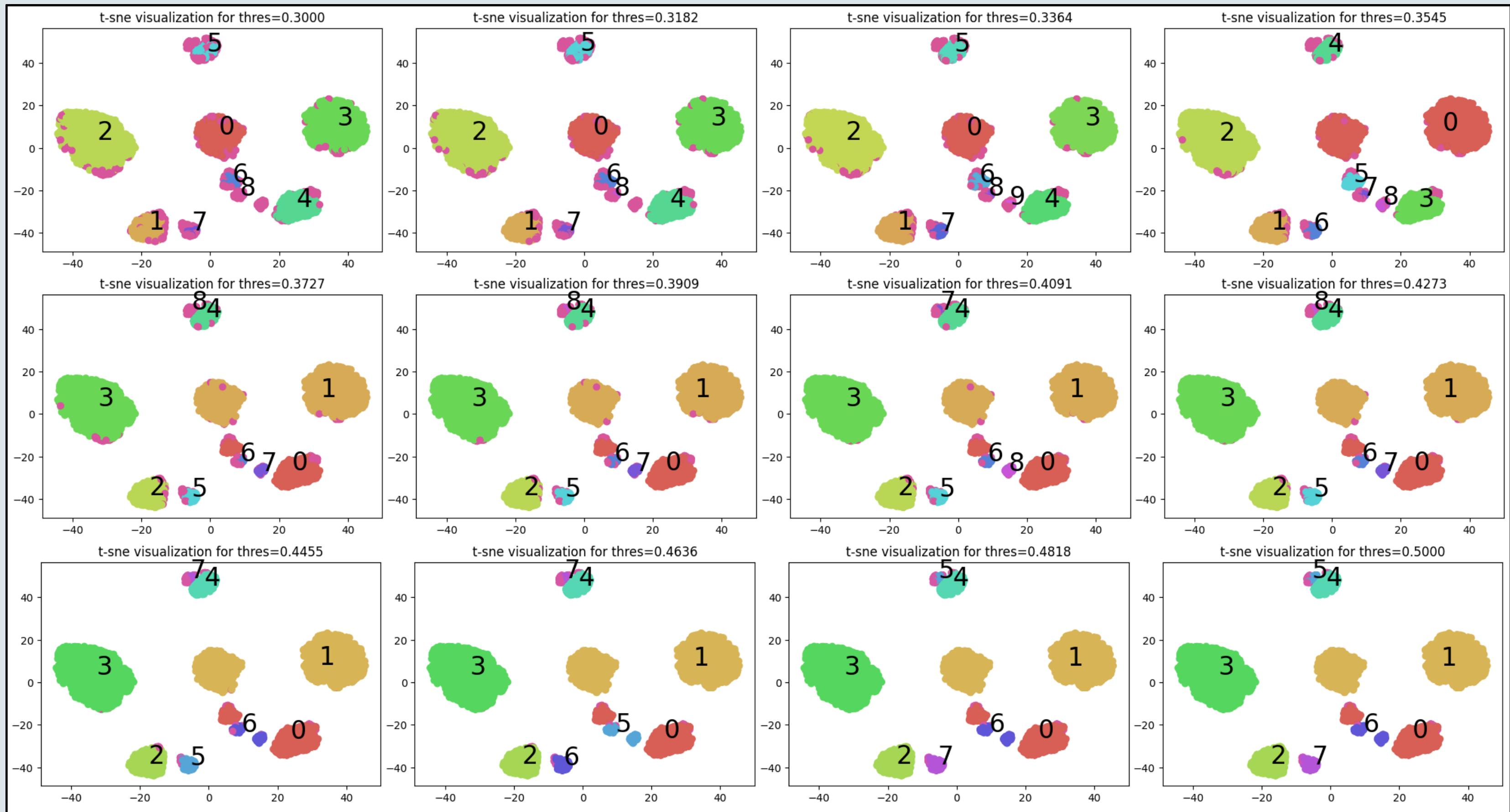
DBSCAN

Xây dựng mô hình



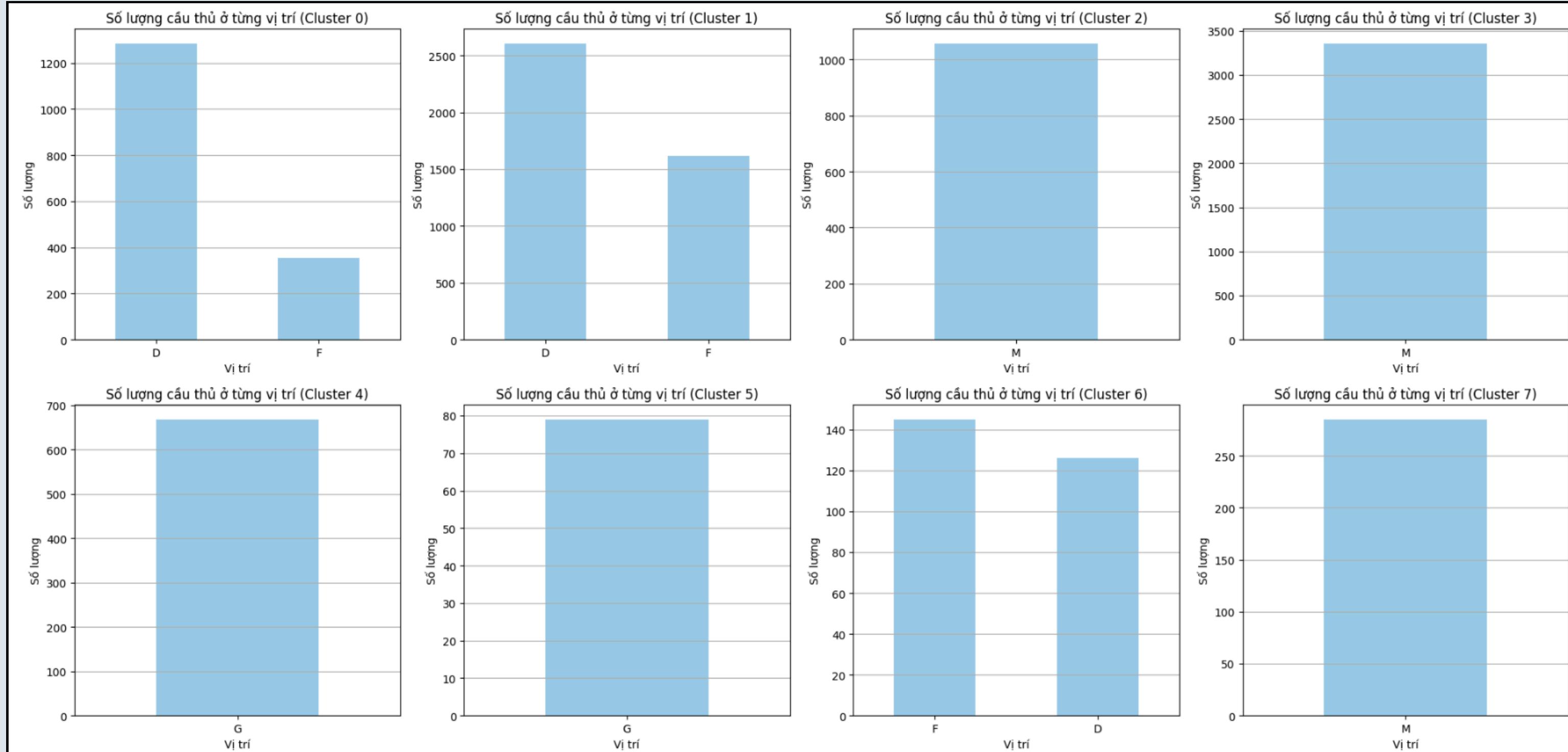
Khoản giá trị của ϵ là từ 2.5 đến 4

DBSCAN



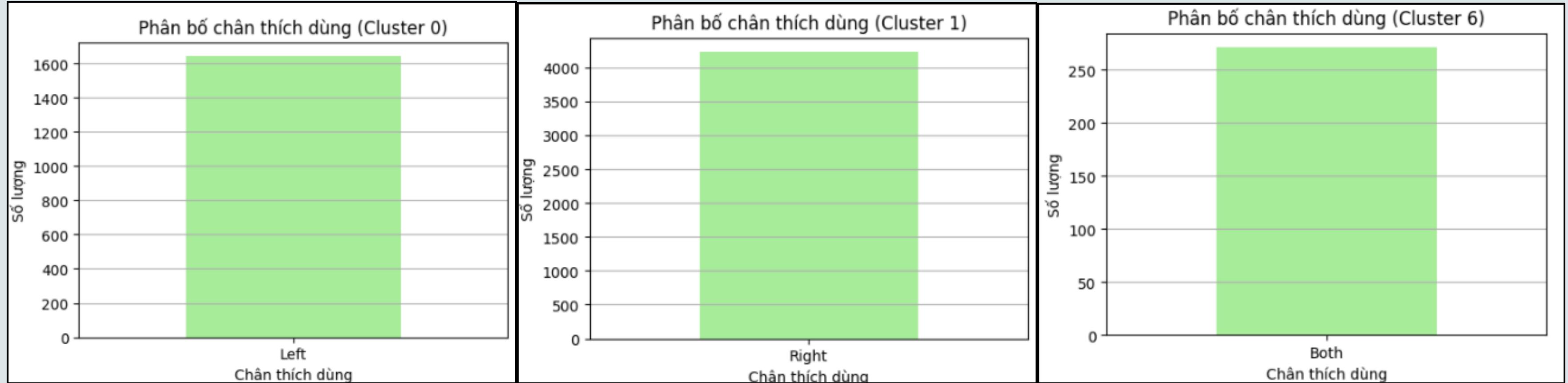
B. DBSCAN

Xây dựng mô hình



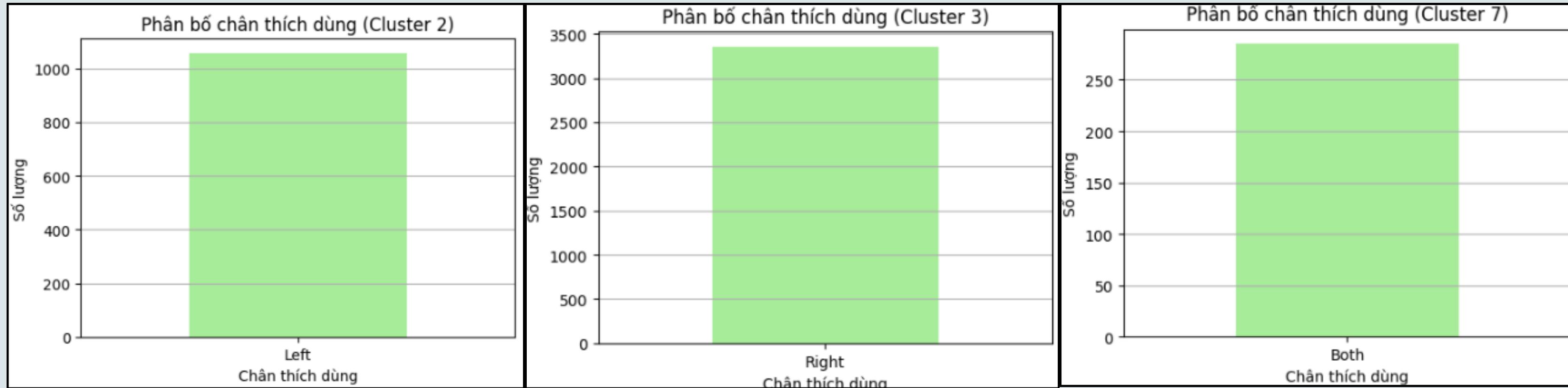
Thống kê vị trí cầu thủ ở từng cụm

B. DBSCAN



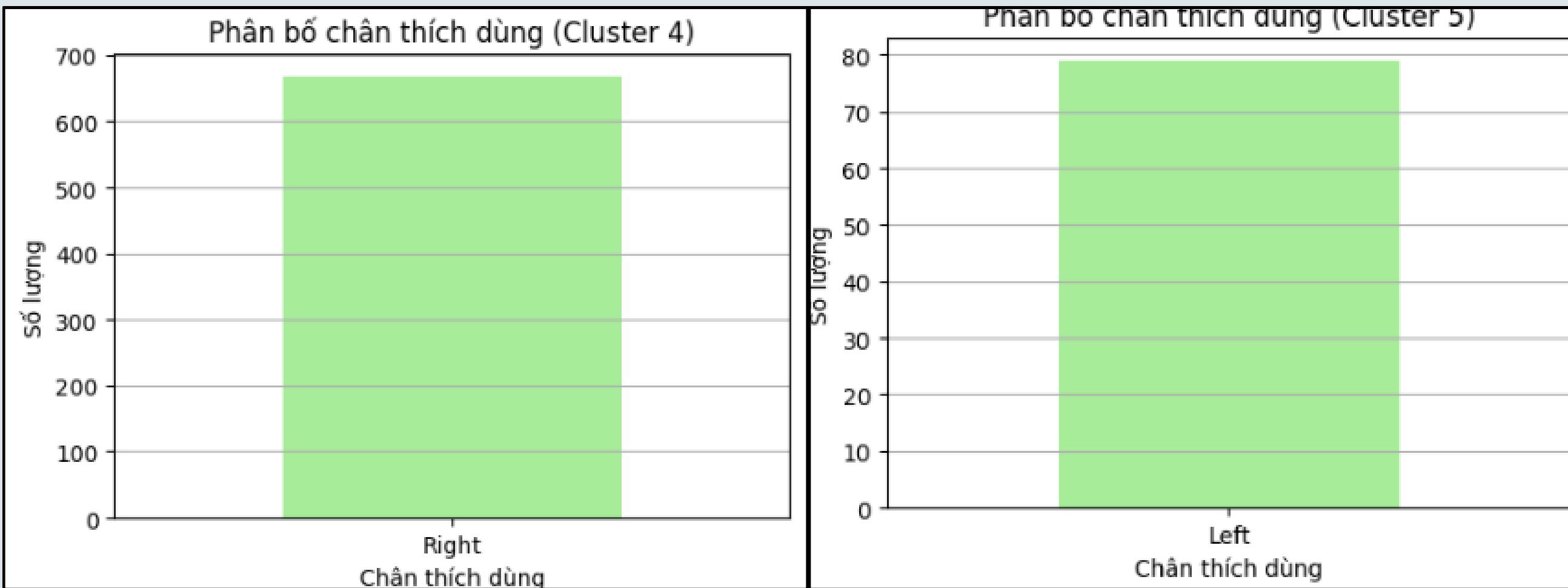
- Các cụm về Hậu vệ (D) và Tiền vệ(F):
 - Cầu thủ cụm 0 sở trường đá chân Trái.
 - Cầu thủ cụm 1 sở trường đá chân Phải.
 - Cầu thủ cụm 6 sở trường cả 2 chân.

B. DBSCAN



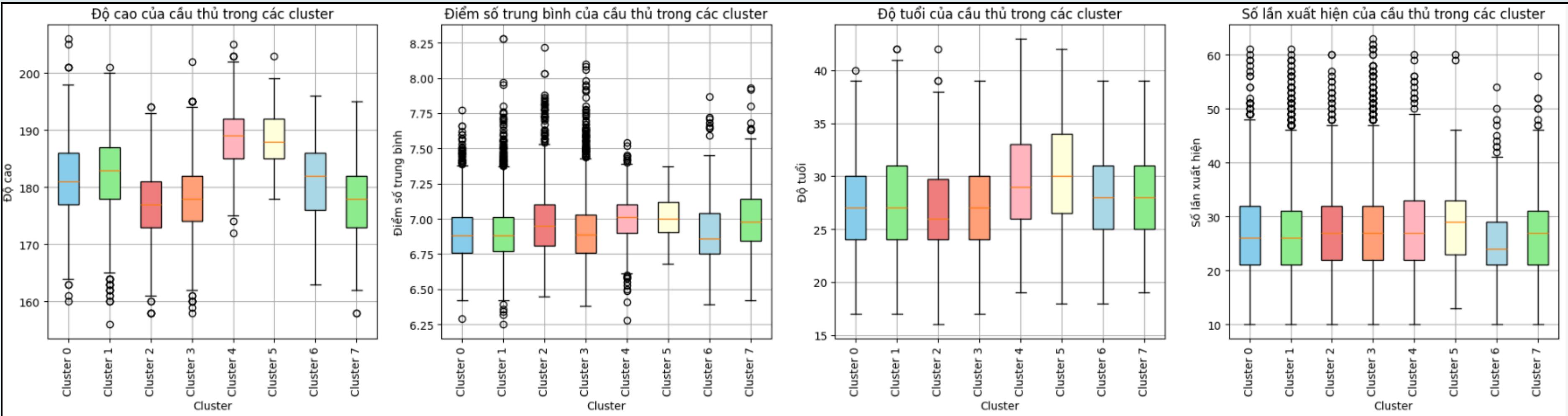
- Cụm về Tiền vệ(M):
 - Tiền vệ cụm 2 sở trường đá chân Trái.
 - Tiền vệ cụm 3 sở trường đá chân Phải.
 - Tiền vệ cụm 7 sở trường cả 2 chân.

B. DBSCAN



- Cụm về Thủ môn(G):
 - Thủ môn cụm 4 sở trường đá chân Phải.
 - Thủ môn cụm 5 sở trường đá chân Trái.

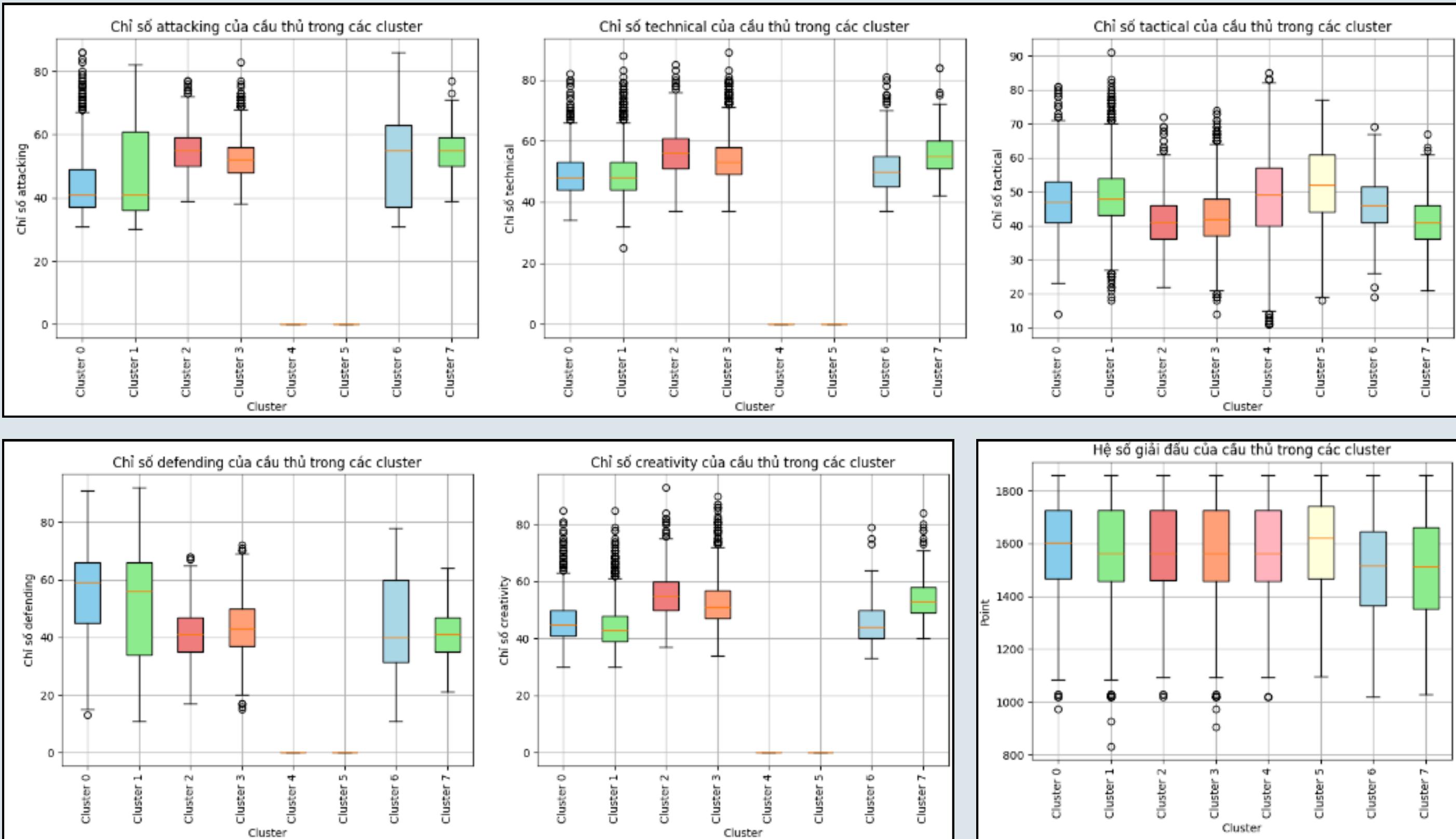
B. DBSCAN



- Các cụm về Tiền đạo(F) và Hậu vệ(D): các chỉ số khá tương đồng nhau
- Các cụm về Tiền vệ(M): các chỉ số khá tương đồng nhau
- Cụm về Thủ môn(G): đặc trưng của vai trò thủ môn khi có Chiều cao và độ tuổi trung bình lớn

Nhận xét:

B. DBSCAN

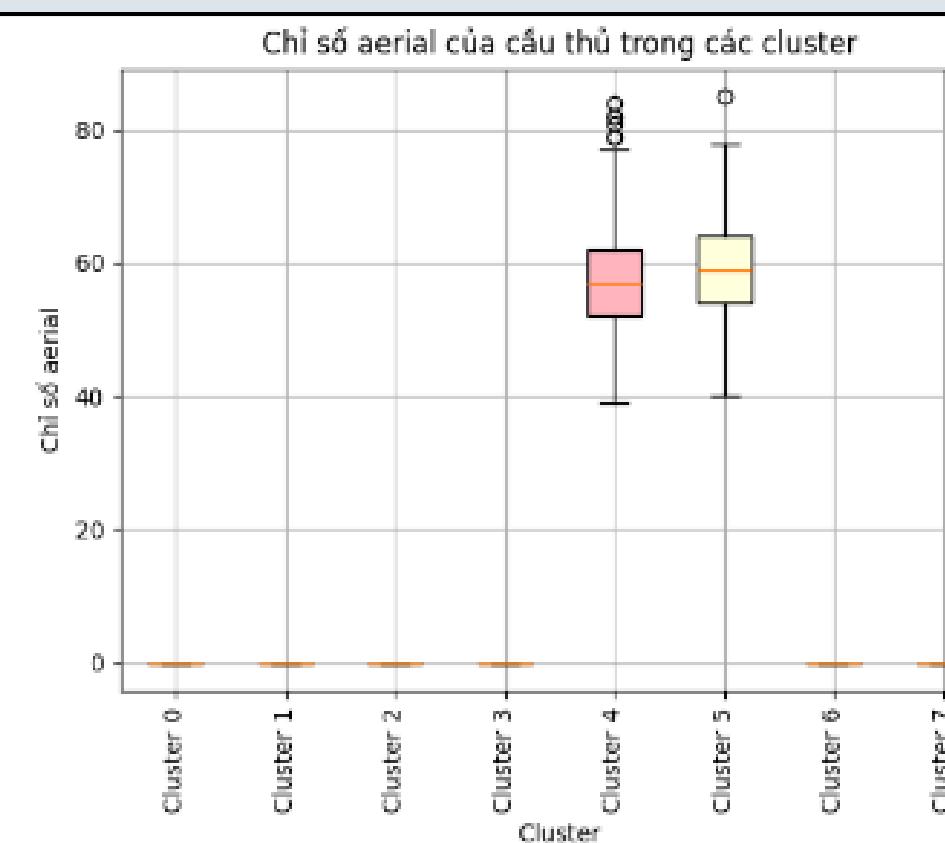
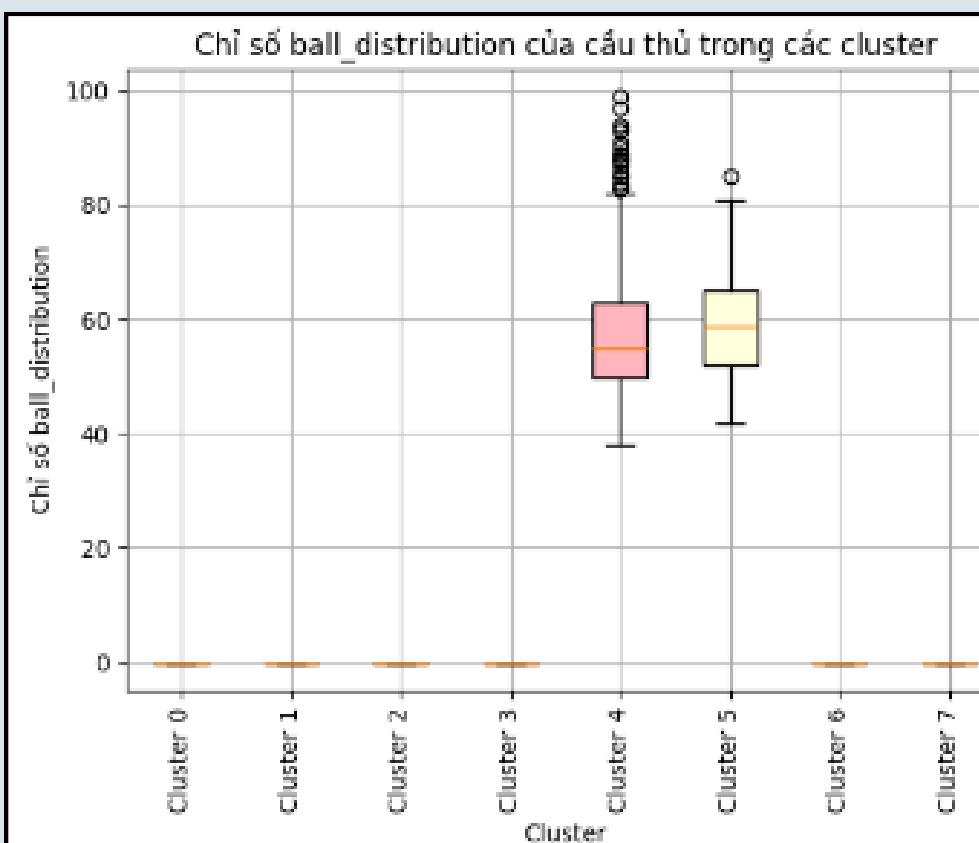
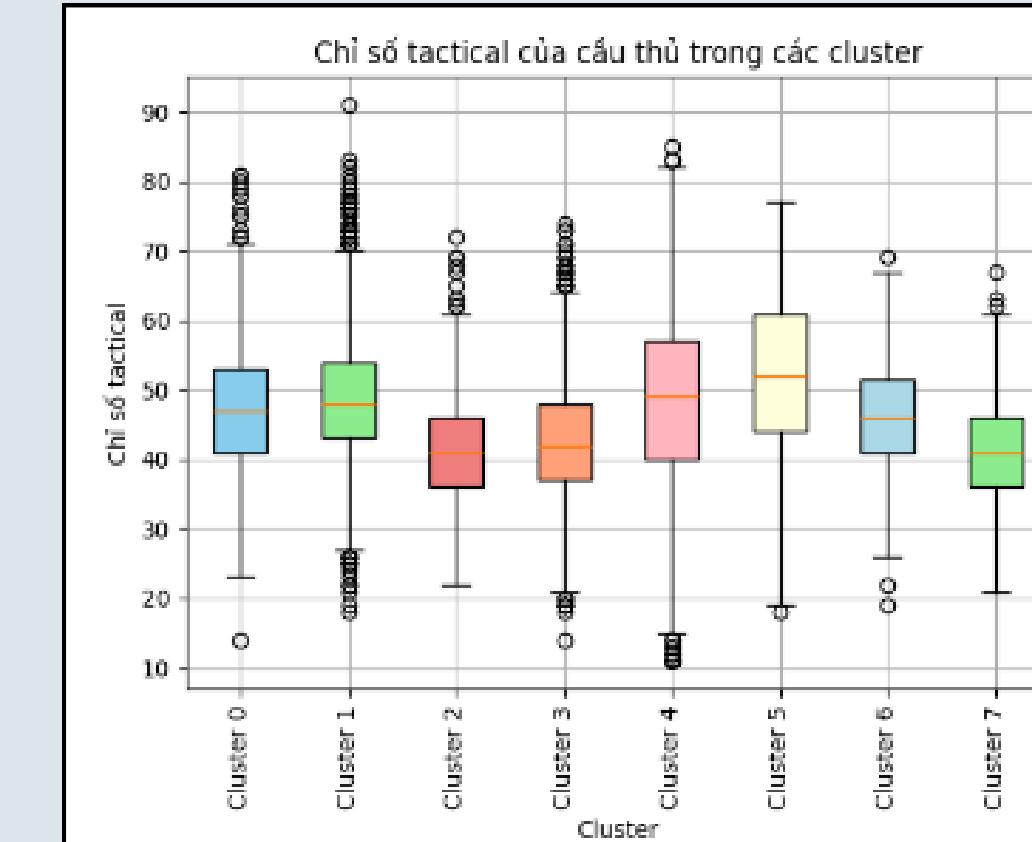
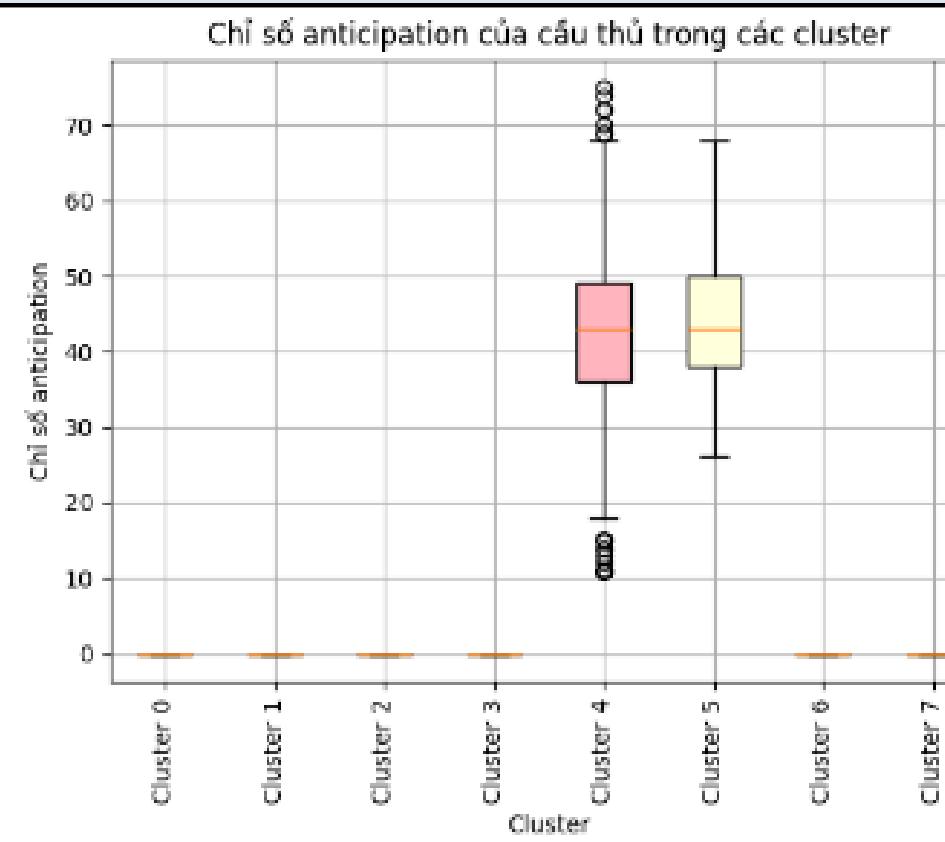
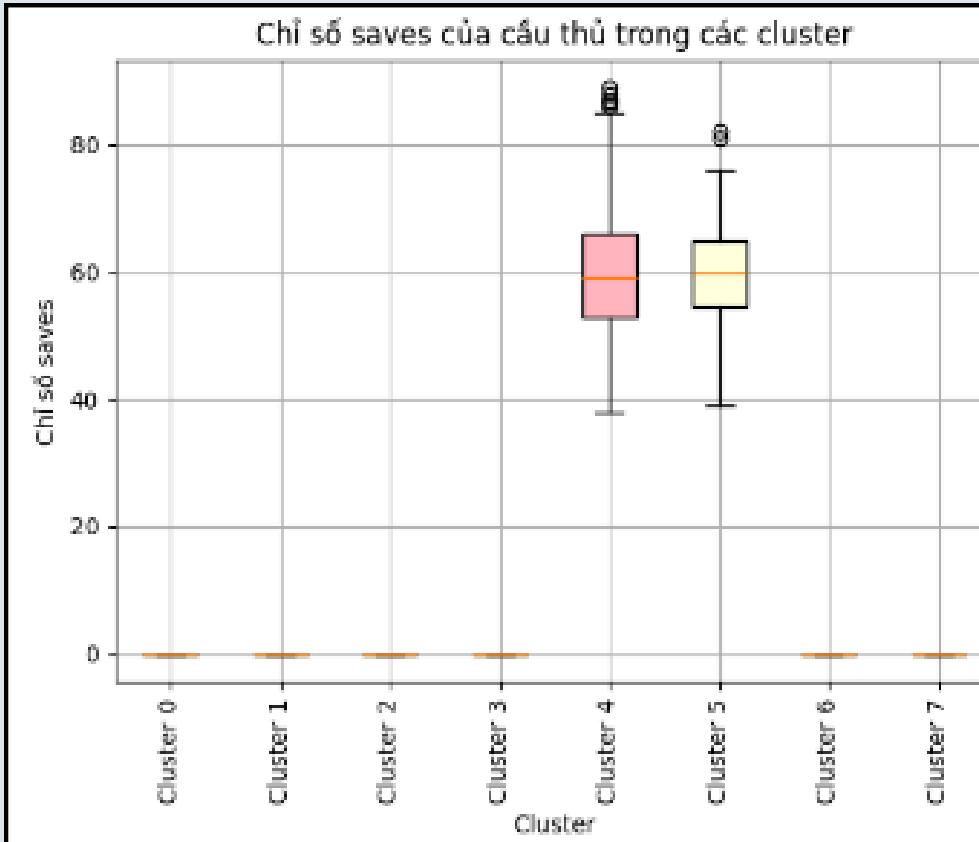


B. DBSCAN

Nhận xét:

- Các cụm về Hậu vệ (D) và Tiền vệ(F): các cầu thủ trong các cụm 0, 1 và 6 có các chỉ số khá tương đồng nhau. Riêng các cầu thủ ở cụm 1 và 6 có chỉ số Attacking cao hơn cụm còn lại, Các cầu thủ ở cụm 0 và 1 có chỉ số Defending cao hơn cụm còn lại. Cụm 6 là các cầu thủ thi đấu ở giải đấu có hệ số thấp hơn
- Cụm về Tiền vệ(M): Các cầu thủ trong các cụm 2, 3 và 7 có các chỉ số khá tương đồng nhau. Riêng cầu thủ ở cụm 2 và 7 có chỉ số Attacking, Technical và Creatitivy cao hơn đôi chút với cụm còn lại, ngược lại cụm 3 lại có chỉ số Tactical và Defending cao hơn. Cụm 7 là các cầu thủ thi đấu ở giải đấu có hệ số thấp hơn

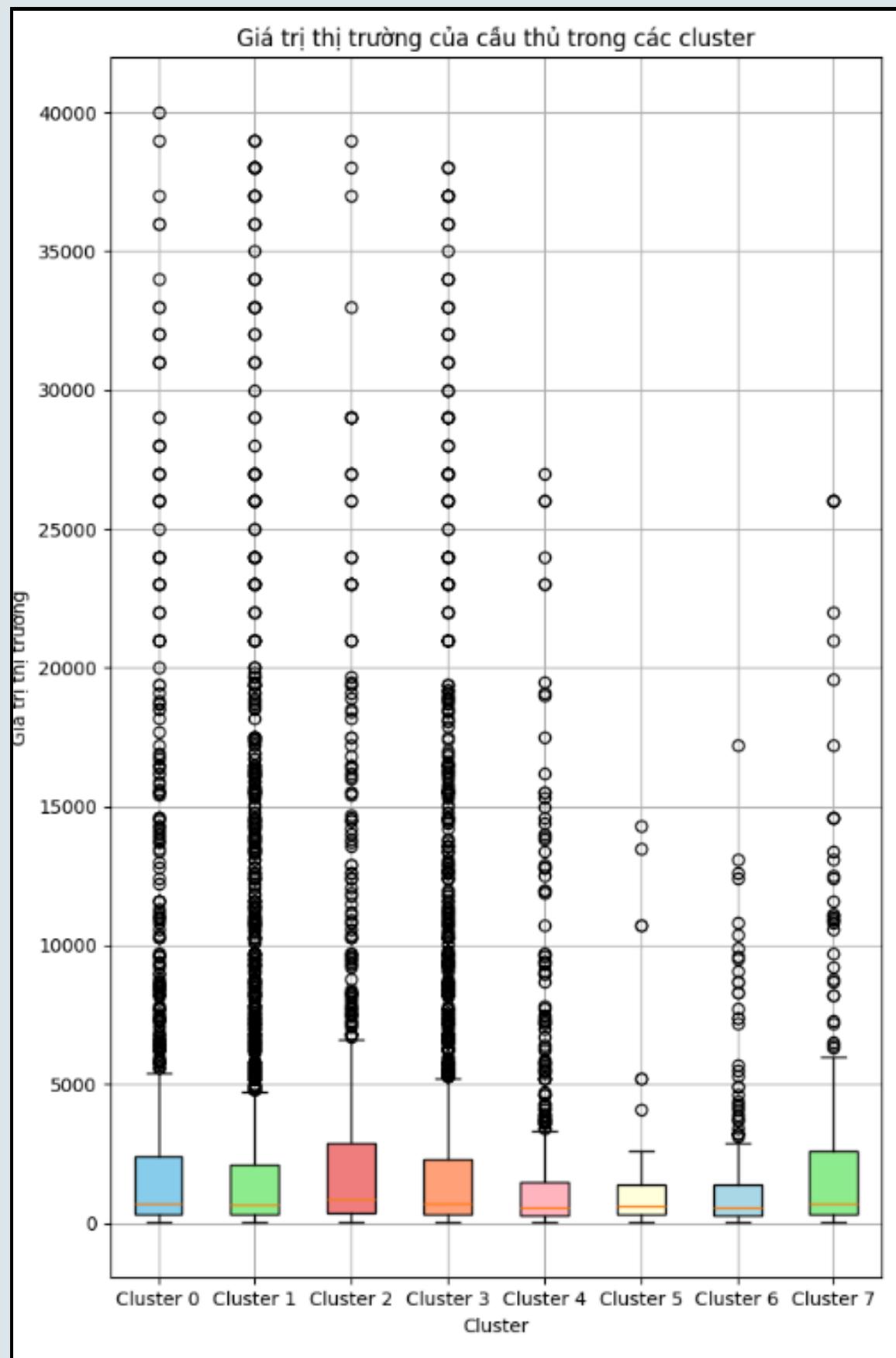
B. DBSCAN



Nhận xét:

Cụm về Thủ môn(G): Cụm 5 là các thủ môn có chỉ số Tactical, Anticipation, Ball_distribution và Aerial cao hơn so với cụm 4, tuy nhiên chỉ số Saves lại kém hơn.

B. DBSCAN



Nhận xét:

- Các cụm về Hậu vệ (D) và Tiền vệ(F): các cầu thủ trong cụm 0 có giá trị trung bình cao nhất trong các cụm Tiền đạo và Hậu vệ, Cụm 6 có giá trị thị trường trung bình thấp nhất
- Cụm về Tiền vệ (M): Các tiền vệ trong cụm 2 có giá trị thị trường trung bình cao nhất, tiền vệ cụm 3 giá trị thị trường trung bình thấp nhưng có nhiều cá biệt có giá trị thị trường cao đáng kể
- Cụm về Thủ môn(G): Các thủ môn trong cụm 4 có giá trị thị trường cao hơn cụm 5 và cũng có nhiều cầu thủ cá biệt có giá trị thị trường cao đáng kể.

B. DBSCAN

Kết luận

- Khi sử dụng epsilon = 0.4818 và minPts = 34, DBSCAN phân dữ liệu thành 8 cụm.
- Phương pháp phân cụm dựa trên mật độ Density-Based Clustering cho ra kết quả các cụm có một số tính chất khác và cũng có một số tính chất mà một số cụm có sự tương đồng về giá trị biến dữ liệu tùy theo việc định tham số minPts và Epsilon
- DBSCAN cho ra kết quả phân cụm không thể hiển tốt mối quan hệ giữa các cụm.
- Các cụm thu đờng:
 - Cụm 0: Là Tiền đạo/Tiền vệ, defending khá cao, chân trái, giá trị trung bình cao
 - Cụm 1: Là Tiền đạo/Tiền vệ, attacking|defending khá cao, chân phải
 - Cụm 2: Là Tiền vệ, attacking|technical|creativity trung bình cao, chân trái, giá trị trung bình cao
 - Cụm 3: Là Tiền vệ, defending|tactical trung bình cao, chỉ số cao toàn diện, chân phải, giá trị trung bình thấp
 - Cụm 4: Là Thủ môn, saves cao, chân phải, giá trị trung bình cao
 - Cụm 5: Là Thủ môn, tactical|anticipation|ball_distribution|aerial cao, saves trung bình, chân trái
 - Cụm 6: Là Tiền đạo/Tiền vệ, attacking khá cao, giá đấu nhỏ, thuận 2 chân, giá trị trung bình thấp
 - Cụm 7: Là Tiền vệ, attacking|technical|creativity trung bình cao, giải đấu nhỏ, thuận 2 chân

4. Kết luận, đánh giá chung

• • • •

- Cả 2 phương pháp phân cụm Hierarchical Clustering và DBSCAN đều cho ra các cụm có các tính chất khác nhau khi dùng các tham số đầu vào của mô hình một cách thích hợp
- Cả 2 phương pháp đều hỗ trợ phân cụm dữ liệu mà không cần xác định trước số cụm
- Hierarchical Clustering mất thời gian tính toán trên tập dữ liệu lớn nhưng cho ra kết quả phân cụm thể hiện tốt mối quan hệ giữa các cụm. Trong khi DBSCAN có thể xử lý các điểm nhiễu hiệu quả hơn nhưng khó khăn trong xử lý dữ liệu có nhiều điểm lân cận



Lựa chọn phương pháp **Hierarchical Clustering** để phân cụm dữ liệu

• • • •

.....

Cảm ơn các bạn
đã lắng nghe

.....