TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

TÀI LIỆU HƯỚNG DẪN THỰC HÀNH
Môn: Khai thác dữ liệu

BÀI THỰC HÀNH SỐ 5

I. Mục tiêu

1. Hiểu và vận dụng giải thuật *k*-Means để gom cụm dữ liệu.
2. Hiểu và vận dụng giải thuật mạng Kohonen để gom cụm dữ liệu.
3. Hiểu các phương pháp đánh giá kết quả đã gom cụm được.

II. Thời gian

1. Thực hành: 4 tiết
2. Bài tập làm thêm: 8 tiết

III. Hướng dẫn chung

1. Yêu cầu cơ bản

Cho bảng dữ liệu do máy tính thu thập được sau khi phân tích 06 bức tranh như sau:

Tranh	Số màu	Số đường nét	Số hình khối
1	16	124	19
2	6	13	70
3	10	22	59
4	5	81	92
5	21	97	23
6	7	94	88

1. Dựa vào dữ liệu ở trên sinh viên sử dụng thuật toán *k*-Means để gom thành 3 nhóm tranh.
2. Hãy áp dụng thuật toán mạng Kohonen để gom các bức tranh thành 3 nhóm với các thông số như sau: số lần lặp $epochs = 5$, bán kính $R = 0$ và tốc độ học (learning rate) $\alpha = 0{,}4$.
3. So sánh kết quả thu được từ thuật toán *k*-Means và mạng Kohonen với nhau.
4. Một bức tranh chưa biết có kết quả phân tích như sau:

Tranh	Số màu	Số đường nét	Số hình khối
7	13	95	73

Sinh viên hãy giúp máy tính tìm ra những bức tranh có đặc điểm tương đồng với bức tranh trên.

Hướng dẫn:

1. Sinh viên tham khảo các bước thực hiện thuật toán k-Means trong tài liệu[1, 2]. Áp dụng trên dữ liệu đề bài ta có:

   Khởi tạo ma trận phân hoạch

	Tranh 1	**Tranh 2**	**Tranh 3**	**Tranh 4**	**Tranh 5**	**Tranh 6**
Cụm 1	*1*	*0*	*0*	*0*	*0*	*0*
Cụm 2	*0*	*1*	*0*	*0*	*0*	*0*
Cụm 3	*0*	*0*	*1*	*1*	*1*	*1*

   Vector trọng tâm của các cụm lúc này là:

   - Cụm 1: $V1 = (16, 124, 19)$
   - Cụm 2: $V2 = (6, 13, 70)$
   - Cụm 3: $V3 = (10.75, 73.5, 65.5)$

   Khoảng cách Euclide từ các bức tranh đến lần lượt các cụm là:

	Cụm 1	**Cụm 2**	**Cụm 3**
Tranh 1	*0*	122.5643	68.84811
Tranh 2	122.5643	*0*	60.85279
Tranh 3	109.7269	*14.76482*	51.91399
Tranh 4	85.43419	71.47727	*28.13472*
Tranh 5	*27.74887*	97.41663	49.63429
Tranh 6	75.77599	82.98193	*30.66859*

   Ma trận phân hoạch các điểm thuộc cụm

	Tranh 1	**Tranh 2**	**Tranh 3**	**Tranh 4**	**Tranh 5**	**Tranh 6**
Cụm 1	*1*	*0*	*0*	*0*	*1*	*0*
Cụm 2	*0*	*1*	*1*	*0*	*0*	*0*
Cụm 3	*0*	*0*	*0*	*1*	*0*	*1*

   Vector trọng tâm của các cụm lúc này là:

   - Cụm 1: $V1 = (18.5\ 110.5, 21)$
   - Cụm 2: $V2 = (8, 17.5, 64.5)$
   - Cụm 3: $V3 = (6, 87.5, 90)$

Khoảng cách Euclide từ các bức tranh đến lần lượt các cụm là:

	Cụm 1	**Cụm 2**	**Cụm 3**
Tranh 1	*13.87444*	116.0883	80.45651
Tranh 2	109.834	*7.382412*	77.13786
Tranh 3	96.68764	*7.382412*	72.57582
Tranh 4	78.06087	69.26399	*6.873864*
Tranh 5	*13.87444*	90.61733	69.3127
Tranh 6	69.95356	80.03437	*6.873864*

Như vậy các điểm thuộc cụm không thay đổi, thuật toán dừng lại, với kết quả sau:

- Tranh 1 & 5 thuộc cụm 1
- Tranh 2 & 3 thuộc cụm 2
- Tranh 4 & 6 thuộc cụm 3

2. Mạng Kohonen (hay còn gọi là SOM) là một mạng nơron truyền thẳng sử dụng phương pháp học không giám sát, áp dụng trong việc ánh xạ để giảm kích thước dữ liệu đầu vào. Từ tập các đối tượng trong không gian nhiều chiều ở đầu vào, mạng Kohonen có số chiều nhỏ hơn (thường là 2 chiều) được sử dụng để đặc trưng cho chúng ở đầu ra của thuật toán.

   Để áp dụng mạng Kohonen vào bài toán gom cụm thì mạng ở đầu ra có số nơron tương ứng với số cụm. Mỗi nơron chứa một vector trọng số có số chiều bằng số chiều của vector dữ liệu đầu vào. Cụ thể như sau:

   - Đầu vào thuật toán: đối tượng cần gom cụm là tập các vector trong không gian n chiều, số lần lặp của thuật toán *epochs*, bán kính vùng lân cận R và tốc độ học α.
   - Đầu ra thuật toán: bản đồ mạng Kohonen với mỗi nơron trên mạng đặc trưng cho một cụm.

   Thuật toán trải qua các bước như sau:

Bước	**Thao tác**
0	Khởi tạo giá trị của các vector trọng số. Gán giá trị cho biến R và α.
1	Nếu chưa thỏa điều kiện dừng thì lặp lại từ bước 2 đến bước 8
2	Với mỗi vector đầu vào x thực hiện bước 3 đến bước 5

3	Với mỗi nơron trên mạng j, tính khoảng cách Euclide đến x theo công thức: $$D(j) = \sqrt{\sum_{i=1}^{n}(x_i - w_{ij})^2}$$
4	Tìm nơron J sao cho khoảng cách từ điểm dữ liệu đang xét x đến J là ngắn nhất.
5	Duyệt qua tất cả những vector trọng số thuộc nơron láng giềng j của J và xét tất cả các chiều i, tiến hành cập nhật $$w_{ij}(new) = w_{ij}(old) + \alpha(x_i - w_{ij}(old))$$
6	Cập nhật lại tốc độ học α. Điều này giúp làm giảm số lần lặp.
7	Giảm bán kính vùng lân cận vào thời điểm thích hợp
8	Kiểm tra điều kiện dừng. Thông thường thì đó là lúc giá trị của tốc độ học đã rất nhỏ hoặc giá trị của vector trọng số hội tụ nên việc cập nhật không còn thay đổi nhiều.

Với bài tập trên, trước hết cần xác định các tham số của thuật toán:

- Dữ liệu đầu vào là thông tin của 06 bức tranh, mỗi thông tin có thể xem như là một vector trong không gian $n = 3$ chiều.
- Bán kính vùng lân cận là $R = 0$, điều này có nghĩa là khi thay đổi trọng số của một nơron thì những nơron lân cận không bị ảnh hưởng.
- Số lần lặp của thuật toán là $epochs = 5$
- Tốc độ học là $\alpha = 0.4$
- Vì mục đích của bài toán là gom thành 3 cụm nên mạng Kohonen ở đầu ra gồm 3 nơron. Mỗi nơron chứa 1 trọng số là vector w_i có số chiều bằng n.

Khởi tạo giá trị của các vector trọng số:

$$w_1 = (19, 111, 21.5)$$
$$w_2 = (6.5, 88, 90.5)$$
$$w_3 = (8.5, 18, 65)$$

Ở lần lặp thứ nhất:

- Xét vector đầu tiên (Tranh 1) x_1

  Khoảng cách từ x_1 đến w_1 là:

$$D_{11} = \sqrt{(16-19)^2 + (124-111)^2 + (19-21.5)^2} \approx 13.57$$

Tương tự, khoảng cách từ x_1 đến w_2, w_3 lần lượt là: $D_{12} \approx 80.61, D_{13} \approx 115.79$

Như vậy, nơron có trọng số w_1 là nơron có khoảng cách đến x_1 ngắn nhất. Tiến hành cập nhật lại trọng số của w_1

$$w_{11}(new) = 19 + 0.4 \times (16 - 19) = 17.8$$
$$w_{12}(new) = 111 + 0.4 \times (124 - 111) = 116.2$$
$$w_{13}(new) = 21.5 + 0.4 \times (19 - 21.5) = 20.5$$

Lúc này, do $R = 0$ nên không cần cập nhật lại vùng lân cận, w_2, w_3 sẽ giữ nguyên giá trị, ta có:

$$w_1 = (17.8, 116.2, 20.5)$$
$$w_2 = (6.5, 88, 90.5)$$
$$w_3 = (8.5, 18, 65)$$

- Lần lượt xét các vector còn lại

  Tương tự như trên, sau khi xét lần lượt các vector còn lại, ta có trọng số lúc này là:

$$w_1 = (18.72, 110.08, 21.2)$$
$$w_2 = (6.34, 88.72, 89.86)$$
$$w_3 = (8.5, 18.4, 63.8)$$

- Trước khi kết thúc lần lặp đầu tiên, ta giảm tốc độ học đi một nửa:

$$\alpha = \frac{0.4}{2} = 0.2$$

Tiếp tục lặp lại các thao tác trên thêm 4 lần, ta được:

$$w_1 = (18.7, 109.78, 21.17)$$
$$w_2 = (6.18, 88.19, 89.9)$$
$$w_3 = (8.32, 18.1, 63.95)$$

Tính khoảng cách từ mỗi điểm đến các vector ta kết luận được:

- Tranh 1 & 5 thuộc cụm đại diện bởi vector 1
- Tranh 4 & 6 thuộc cụm đại diện bởi vector 2
- Tranh 2 & 3 thuộc cụm đại diện bởi vector 3

3. Từ kết quả thu được ở câu 1 và 2, có thể thấy ở cả hai thuật toán các bức tranh trong một cụm đều giống nhau. Các cụm tương ứng ở hai thuật toán là:

k-Means	Mạng Kohonen
Cụm 1	*Cụm 1*
Cụm 2	*Cụm 3*
Cụm 3	*Cụm 2*

4. Sinh viên có thể tìm ra sự tương đồng bằng cách áp dụng kết quả của thuật toán *k*-Means hoặc mạng Kohonen. Nếu áp dụng kết quả gom cụm từ thuật toán *k*-Means, chúng ta sẽ phải tính khoảng cách Euclide từ điểm đại diện cho Tranh 7 đến tâm các cụm. Sau đó chọn ra cụm có khoảng cách gần nhất, đó chính là cụm chứa các bức tranh có sự tương đồng nhiều nhất với Tranh 7. Tương tự như trên, nếu áp dụng kết quả từ mạng Kohonen, chúng ta sẽ phải tính khoảng cách Euclide từ điểm đại diện cho Tranh 7 đến vector trọng số của các nơron.

   Đối với thuật toán *k*-Means:

   - Khoảng cách đến cụm 1

     $$D_{71} = \sqrt{(13 - 18.7)^2 + (95 - 110.5)^2 + (73 - 21)^2} \approx 54.54$$

   - Tương tự, khoảng cách đến cụm 2 và 3 lần lượt là $D_{72} \approx 78.12$; $D_{73} \approx 19.86$
   - Vì khoảng cách từ Tranh 7 tới tâm cụm 3 là ngắn nhất nên có thể kết luận được rằng Tranh 7 có sự tương đồng với các bức tranh 4 và 6.

   Đối với thuật toán mạng Kohonen:

   - Khoảng cách đến nơron 1

     $$D_{71} = \sqrt{(13 - 18.5)^2 + (95 - 109.78)^2 + (73 - 21.17)^2} \approx 54.2$$

   - Tương tự, khoảng cách đến nơron 2 và 3 lần lượt là $D_{72} \approx 19.46$; $D_{73} \approx 77.57$
   - Vì khoảng cách từ Tranh 7 tới nơron 2 là ngắn nhất nên có thể kết luận được rằng Tranh 7 có sự tương đồng với các bức tranh 4 và 6.

   Như vậy, dù với phương pháp nào thì chúng ta cũng có cùng một kết luận về sự tương đồng của Tranh 7 với tranh 4 và 6.

2. Yêu cầu lập trình

Cho dữ liệu Mua sắm của khách hàng Mall customer[3] bao gồm các thông tin sau: Giới tính (Gender), Tuổi (Age), Thu nhập hằng năm (Annual income – đơn vị tính: nghìn đô)

và Số điểm mua sắm (Spending Score – từ 1 đến 100). Sinh viên hãy thực hiện những yêu cầu sau đây:

1. Nhập dữ liệu, thống kê các thông tin cơ bản, xử lý dữ liệu bị thiếu.
2. Sử dụng biểu đồ tần suất (Histogram) để biểu diễn phân phối của lần lượt các thuộc tính Tuổi, Thu nhập hằng năm và Số điểm mua sắm.
3. Thống kê số mẫu dữ liệu theo thuộc tính Giới tính.
4. Sử dụng biểu đồ phân tán (Scatter plot) để khảo sát phân bố của thuộc tính Tuổi và Thu nhập hằng năm theo Giới tính.
5. Tương tự câu 4, khảo sát thuộc tính Thu nhập hằng năm và Số điểm mua sắm theo Giới tính.
6. Chọn thuộc tính Tuổi và Số điểm mua sắm để gom cụm. Sử dụng phương pháp khuỷu tay (ELBOW method) để xác định số cụm cần thiết. Tiến hành gom cụm bằng thuật toán *k*-Means với số cụm vừa xác định.
7. Biểu diễn kết quả gom cụm trên bằng biểu đồ.
8. Đánh giá kết quả gom cụm bằng Hệ số Dáng điệu (Silhouette coefficient).
9. Thực hiện tương tự câu 6-8 với trường hợp gom cụm theo các thuộc tính:
   - Thu nhập hằng năm và Số điểm mua sắm.
   - Tuổi, Thu nhập hằng năm và Số điểm mua sắm (thực hiện vẽ biểu đồ 3D để biểu diễn kết quả gom cụm).

Hướng dẫn

Để thực hiện biểu diễn kết quả gom cụm 3D trên ngôn ngữ Python, sinh viên cần cài đặt thư viện *plotly*.

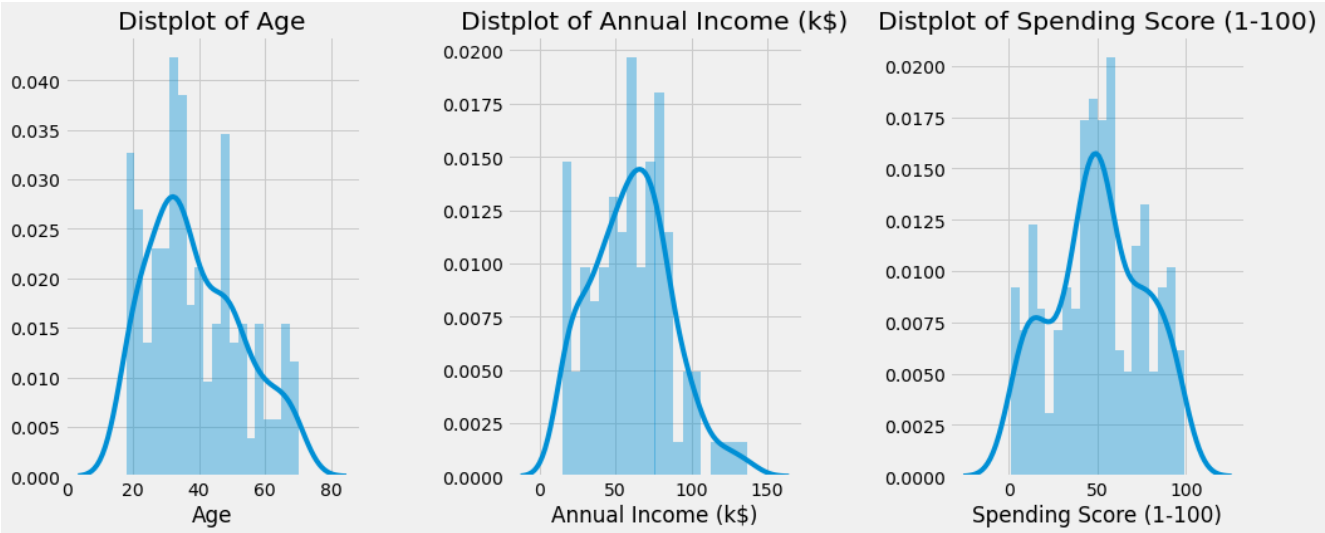
Sau đó, thực hiện import các thư viện sau vào bài làm.

```
import pandas as pd
import numpy as np

from sklearn.cluster import KMeans
from sklearn import metrics

import seaborn as sns
import plotly as py
import plotly.graph_objs as go
import matplotlib.pyplot as plt
%matplotlib inline
```

1. Sinh viên tự thực hiện các thao tác nhập dữ liệu, thống kê các thông tin cơ bản, xử lý dữ liệu bị thiếu như đã học ở các bài thực hành trước.
2. Sử dụng biểu đồ tần suất (Histogram) trong thư viện matplotlib để biểu diễn phân phối của lần lượt các thuộc tính Tuổi, Thu nhập hằng năm và Số điểm mua sắm.

```
plt.style.use('fivethirtyeight')
plt.figure(1, figsize = (15, 6))
n = 0
for x in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:
    n += 1
    plt.subplot(1, 3, n)
    plt.subplots_adjust(hspace =0.5, wspace = 0.5)
    sns.distplot(data[x], bins = 20)
    plt.title('Distplot of {}'.format(x))
plt.show()
```

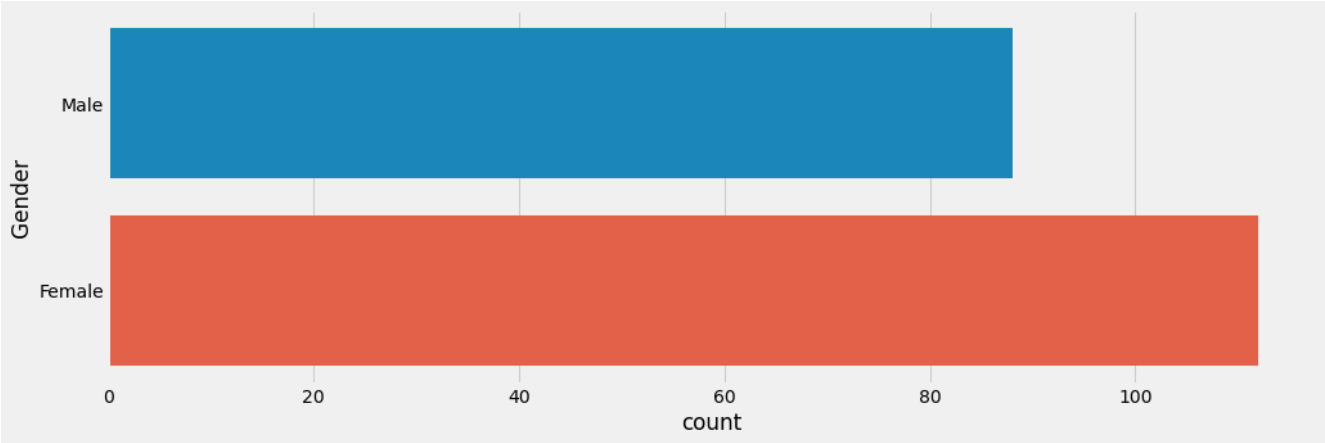
Kết quả thu được là



3. Thống kê số mẫu dữ liệu theo giới tính

```
plt.figure(1, figsize = (15, 5))
sns.countplot(y = 'Gender', data = data)
plt.show()
```

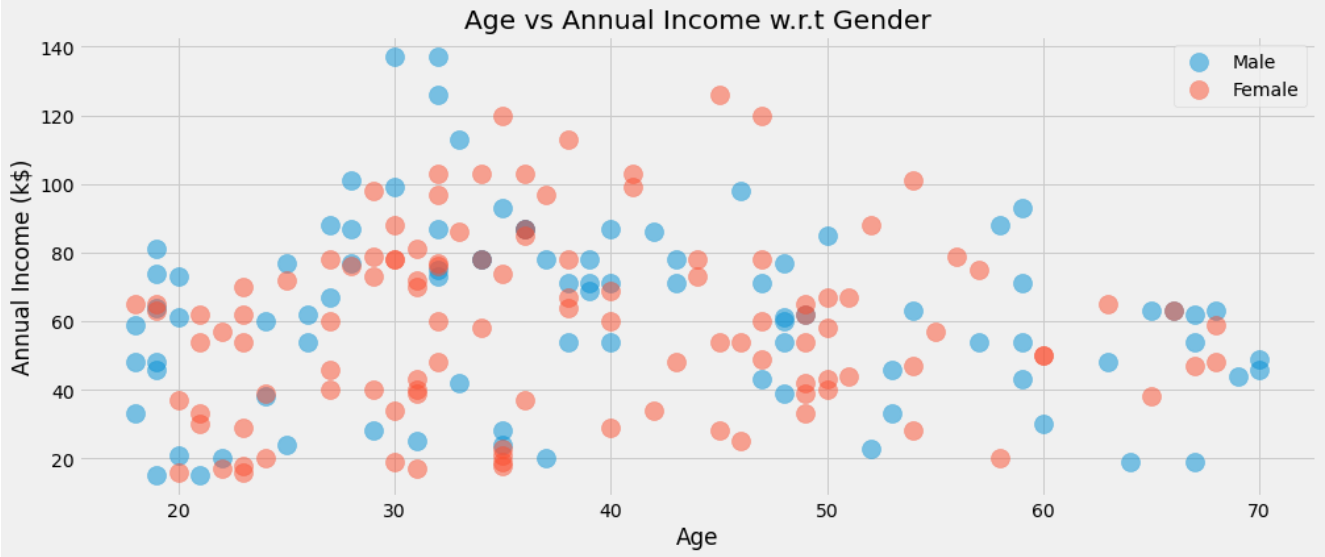
Thu được biểu đồ như sau



4. Sử dụng biểu đồ phân tán (Scatter plot) để khảo sát phân bố của thuộc tính Tuổi và Thu nhập hằng năm theo Giới tính.

```
plt.figure(1, figsize = (15, 6))
for gender in ['Male' , 'Female']:
    plt.scatter(x = 'Age', y = 'Annual Income (k$)', data = data[data['Gender'] == gender],
                s = 200, alpha = 0.5, label = gender)
plt.xlabel('Age'), plt.ylabel('Annual Income (k$)')
plt.title('Age vs Annual Income w.r.t Gender')
plt.legend()
plt.show()
```

Biểu đồ có dạng như sau



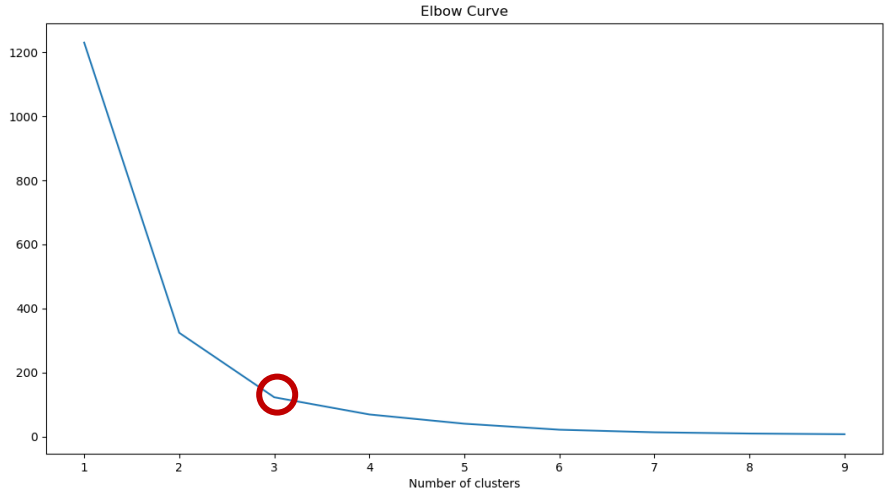
5. Tương tự, sinh viên thực hiện vẽ biểu đồ khảo sát sự phân bố của thuộc tính Thu nhập hằng năm và Số điểm mua sắm theo Giới tính.
6. Sử dụng phương pháp khuỷu tay (ELBOW method) để xác định số cụm cần thiết.

Phương pháp khuỷu tay là một phương pháp heuristic, xem xét k trong một phạm vi nhất định, lần lượt gom cụm theo từng k và tính toán giá trị của tổng bình phương khoảng cách từ các đối tượng thuộc cụm đến tâm cụm WSS (Within-cluster Sum of Square hay còn gọi là Inertia)

$$WSS_k = \sum_{i=0}^{k} \sum_{x \in C_i} d(x,\, c_i)^2$$

Với k là số cụm, x là đối tượng thuộc cụm C_i và c_i là tâm của cụm C_i.

Với kết quả tìm được, thực hiện vẽ đồ thị biểu diễn WSS ứng với mỗi k cụm.



Cuối cùng, chọn số cụm k sao cho khi thêm vào một cụm khác thì không làm cho WSS thay đổi nhiều (vị trí elbow/knee của đường cong)

Khảo sát việc gom cụm theo thuộc tính Tuổi và Số điểm mua sắm

```
X1 = data[['Age' , 'Spending Score (1-100)']].iloc[: , :].values
inertia = []
for n in range(1, 11):
    algorithm = KMeans(n_clusters = n, init='random')
    algorithm.fit(X1)
    inertia.append(algorithm.inertia_)
```

Vẽ biểu đồ biểu diễn giá trị Inertia, theo từng cách gom cụm k

```
plt.figure(1, figsize = (15, 6))
plt.plot(np.arange(1 , 11), inertia, 'o')
plt.plot(np.arange(1 , 11), inertia, '-', alpha = 0.5)
plt.xlabel('Number of Clusters'), plt.ylabel('Inertia')
plt.show()
```

Thu được biểu đồ như sau

Từ biểu đồ trên có thể xác nhận, số cụm tối ưu là 4 (k=4).

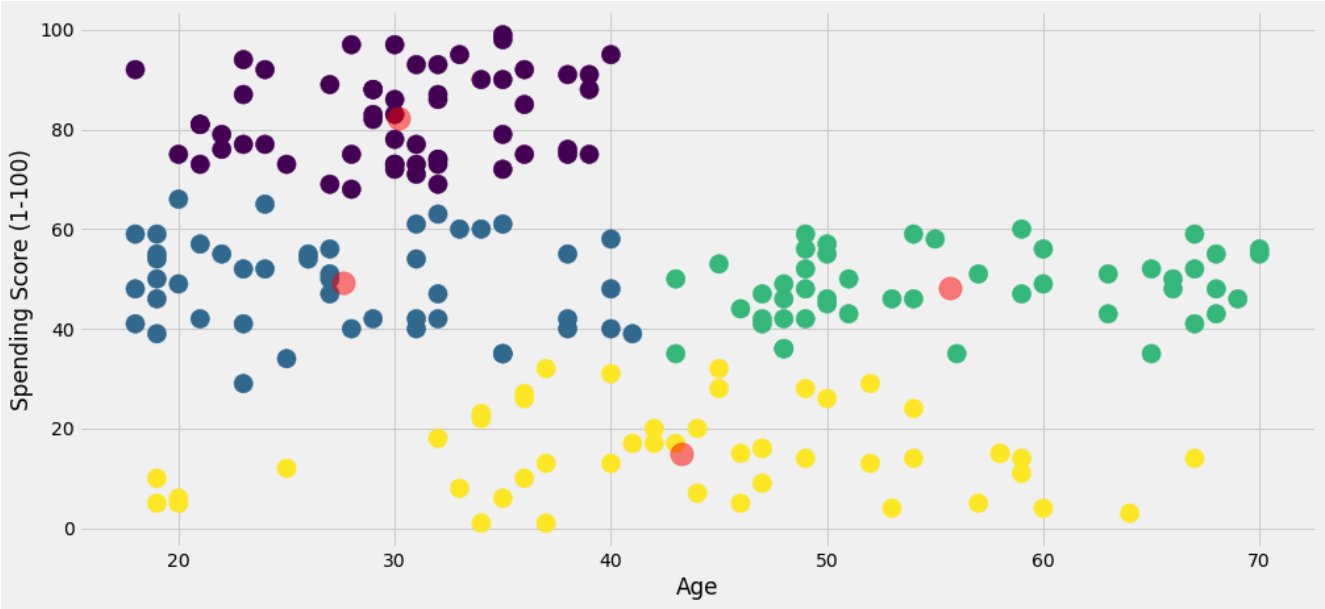
Tiến hành gom cụm dữ liệu Tuổi và Số điểm mua sắm

```
algorithm = KMeans(n_clusters = 4, init='random')
algorithm.fit(X1)
labels1 = algorithm.labels_
centroids1 = algorithm.cluster_centers_
```

7. Biểu diễn kết quả gom cụm bằng biểu đồ

```
plt.figure(1 , figsize = (15, 7) )
plt.clf()
plt.scatter(x = 'Age', y = 'Spending Score (1-100)', data = data, c = labels1,
            s = 200 )
plt.scatter(x = centroids1[: , 0], y =  centroids1[: , 1], s = 300, c = 'red', alpha = 0.5)
plt.ylabel('Spending Score (1-100)'), plt.xlabel('Age')
plt.show()
```

Kết quả gom cụm như sau



8. Đánh giá kết quả gom cụm bằng Hệ số Dáng điệu (Silhouette coefficient)

Hệ số dáng diệu kết hợp cả hai yếu tố gắn kết và tách biệt, nhưng dùng cho cả từng điểm cũng như từng cụm và tổng thể.

Cho một điểm i

- Tính a = trung bình khoảng cách từ điểm i tới tất cả các điểm khác trong cụm.
- Tính b = MIN(trung bình khoảng cách từ điểm i tới các điểm trong cụm khác)
- Hệ số dáng điệu của một điểm s = (b – a)/MAX(a, b)
- Hệ số này thuộc khoảng [0, 1], càng gần 1 thì chất lượng phân cụm càng tốt.

Có thể tính hệ số dáng điệu trung bình cho một cụm hoặc cả một phân cụm.

Sử dụng thư viện *sklearn* để tính hệ số dáng điệu cho trường hợp trên bằng lệnh sau

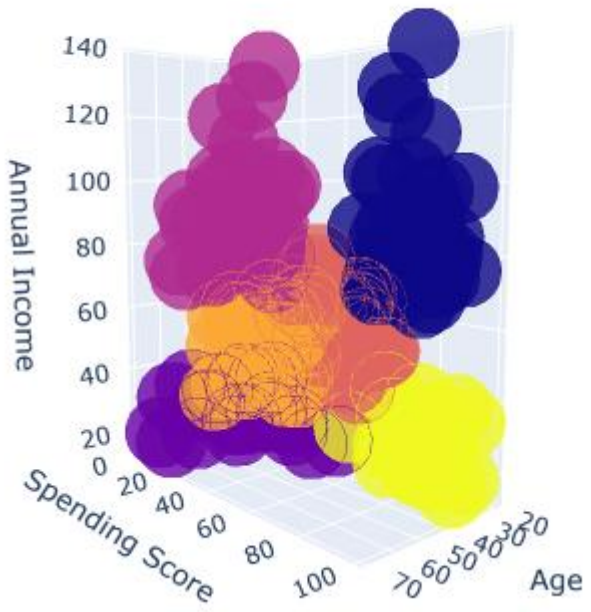
```
metrics.silhouette_score(X1, labels1, metric = 'euclidean')
```

9. Sinh viên thực hiện tương tự với trường hợp gom cụm theo các thuộc tính:
   - Thu nhập hằng năm và Số điểm mua sắm.
   - Tuổi, Thu nhập hằng năm và Số điểm mua sắm.

Thực hiện vẽ biểu đồ 3D, biểu diễn kết quả gom cụm Tuổi, Thu nhập hằng năm và Số điểm mua sắm.

```
data['label3'] =  labels3
trace1 = go.Scatter3d(
    x = data['Age'],
    y = data['Spending Score (1-100)'],
    z = data['Annual Income (k$)'],
    mode='markers',
     marker=dict(
        color = data['label3'],
        size= 20,
        line=dict(
            color= data['label3'],
            width= 12
        ),
        opacity=0.8
     )
)
dat = [trace1]
layout = go.Layout(
    title= 'Clusters',
    scene = dict(
            xaxis = dict(title  = 'Age'),
            yaxis = dict(title  = 'Spending Score'),
            zaxis = dict(title  = 'Annual Income')
        )
)
fig = go.Figure(data=dat, layout=layout)
py.offline.iplot(fig)
```

Kết quả thu được như sau



IV. Thực hành

1. (Cơ bản) Trong quá trình thống kê doanh thu, một công ty kinh doanh chuỗi cửa hàng pizza phát hiện những vị trí có nhu cầu cao nhưng chưa có cửa hàng trong khu vực. Tọa độ tương đối của những vị trí tiềm năng đó được cho trong bảng sau:

Vị trí	*Tọa độ x*	*Tọa độ y*	*Số đơn đặt gần vị trí trong năm*
1	8	4	200
2	8	6	350
3	9	7	650
4	10	5	400
5	11	4	320
6	11	8	250
7	12	6	600
8	12	7	300
9	14	5	200

Công ty này muốn xây dựng 3 cửa hàng pizza mới trong những khu vực trên nhằm mở rộng chuỗi cửa hàng và phục vụ tốt hơn cho những khu vực này. Sinh viên hãy giúp công ty bằng cách tìm ra vị trí đặt 3 cửa hàng sao cho thuận lợi nhất và chỉ ra cụm khách hàng của từng cửa hàng với những yêu cầu cụ thể sau:

a) Sử dụng thuật toán k-Means.

b) Sử dụng mạng Kohonen với các thông số: $epochs = 10$, $R = 0$ và $\alpha = 0{,}8$.

c) So sánh kết quả thu được từ thuật toán k-Means và mạng Kohonen với nhau.

d) Một vị trí tiềm năng mới xuất hiện tại tọa độ:

Vị trí	*Tọa độ x*	*Tọa độ y*	*Số đơn đặt gần vị trí trong năm*
10	11	6	450

Sinh viên hãy giúp công ty xác định cửa hàng nào trong 3 cửa hàng trên sẽ phục vụ cho vị trí này tốt nhất.

2. (Cơ bản) Một website thương mại điện tử chuyên kinh doanh thiết bị gia dụng thu thập được những đơn hàng như sau:

Đơn hàng	Máy lạnh	Máy giặt	Tủ lạnh	Tivi	Bếp điện
1	1	0	1	1	1
2	1	0	1	0	1
3	0	0	1	1	1
4	1	0	0	1	0
5	0	1	0	1	0
6	0	1	0	0	1
7	0	0	0	0	0
8	1	1	1	0	1
9	0	1	0	1	0

Công ty dự định xây dựng một hệ thống khuyến nghị dựa trên việc gom cụm những đơn hàng trên thành 3 nhóm có độ tương đồng với nhau. Từ đó gợi ý cho việc mua hàng của khách hàng dựa trên sự tương đồng này. Sinh viên hãy tham gia xây dựng hệ thống khuyến nghị bằng cách thực hiện những yêu cầu sau:

a) Sử dụng thuật toán k-Means.

b) Sử dụng mạng Kohonen với các thông số: $epochs = 10, R = 0$ và $\alpha = 0{,}8$.

c) So sánh kết quả thu được từ thuật toán k-Means và mạng Kohonen với nhau.

d) Một khách đang thực hiện đặt hàng trên hệ thống đã mua Máy lạnh, tủ lạnh. Theo em những mặt hàng nào nên được khuyến nghị dựa theo kết quả đã tính được từ các câu trên.

3. (Lập trình) Sinh viên thực hiện lại bài tập trong phần Yêu cầu lập trình.
4. (Lập trình) Sinh viên thực hiện gom cụm dữ liệu Titanic[4] bằng thuật toán k-Means và so sánh với thuộc tính phân lớp có sẵn, sử dụng ma trận nhầm lẫn và các độ đo Precision, Recall, F1-score.
5. (Lập trình) Dựa trên dữ liệu MNIST[5] đã được làm quen ở bài thực hành trước. Sinh viên hãy thực hiện thao tác gom cụm bằng thuật toán k-Means. Sau đó so sánh kết quả gom cụm và nhãn lớp có sẵn của dữ liệu này.
6. (Lập trình) Sử dụng thuật toán k-Means để giảm số màu trong ảnh từ đó giảm kích thước file ảnh theo ý muốn. Sinh viên thực hiện theo hướng dẫn sau:

   Khai báo các thư viện vào bài làm

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from PIL import Image
import numpy as np
%matplotlib inline
```

Đọc vào file ảnh cần giảm kích thước và hiển thị ra màn hình

```
image = Image.open('dog.png')
axes = plt.axes(xticks=[], yticks=[])
axes.imshow(image)
```

Chuyển ảnh đọc được thành mảng số và lấy thông tin chiều rộng, cao của ảnh

```
imarr = np.array(image)
rows = imarr.shape[0]
cols = imarr.shape[1]
```

Biến đổi các giá trị màu từ đoạn [0...255] về đoạn [0...1] và chuyển dạng mảng từ 3 chiều (cao, rộng, sâu) thành 2 chiều (cao x rộng, sâu).

```
data = imarr / 255.0 # use 0...1 scale
data = data.reshape(rows * cols, 3)
data.shape
```

```
(405000, 3)
```

Chạy thuật toán gom cụm với số màu mong muốn là 32. Tiếp theo đó, lấy ra mảng biểu diễn hình sau khi đã gom cụm.

```
kmeans = KMeans(n_clusters = 32, max_iter=300)
kmeans.fit(data)
new_colors = kmeans.cluster_centers_[kmeans.predict(data)]
```

Khôi phục dạng mảng vừa thu được từ 2 chiều thành 3 chiều như ban đầu

```
img_recolored = new_colors.reshape(imarr.shape)
```

Biểu diễn hình kết quả và so sánh với ảnh gốc ban đầu

```
fig, ax = plt.subplots(1, 2, figsize=(16, 6),
                       subplot_kw=dict(xticks=[], yticks=[]))
fig.subplots_adjust(wspace=0.05)
ax[0].imshow(image)
ax[0].set_title('Original Image', size=16)
ax[1].imshow(img_recolored)
ax[1].set_title('32-color Image', size=16);
```

Kết quả thu được

Lưu hình kết quả xuống đĩa cứng

```
img_recolored = img_recolored * 255
new_img = Image.fromarray(img_recolored.astype(np.uint8))
new_img.save('compressed_img2.png')
```

V. Bài tập thêm

1. Chọn một ngôn ngữ lập trình, sinh viên hãy cài đặt:

   a) Thuật toán *k*-Medoids

   b) Thuật toán mạng Kohonen

   c) Thuật toán *k*-Means với phương pháp chọn tâm cụm *k*-Means++
2. Có nhiều thuật toán gom cụm tự động xác định được số lượng cụm mà không cần người dùng phải cho trước như *k*-Means. Nhiều người cho rằng đây là điểm vượt trội của chúng so với *k*-Means. Em hãy liệt kê ít nhất 2 trường hợp để phản biện lại quan điểm này.

3. Bạn được cho một tập dữ liệu gồm 100 dòng và yêu cầu gom cụm chúng. Bạn sử dụng thuật toán k-Means để giải quyết bài toán, tuy nhiên với tất cả các giá trị k, $1 \leq k \leq 100$, thuật toán k-Means đều cho ra kết quả là một cụm không rỗng duy nhất. Bạn lại tìm cách áp dụng tất cả các thuật toán cải tiến của k-Means và đều nhận được kết quả tương tự. Bạn hãy giải thích vì sao?
4. Tìm hiểu thêm các thuật toán gom cụm phổ biến khác:
   a) k-Modes, CLARANS
   b) Chameleon, BIRCH
   c) DBSCAN, OPTICS
   d) STING, CLIQUE
5. Để kiểm tra xem khối u, tổn thương trong ngực bệnh nhân có phải là ung thư hay không, người ta thực hiện phương pháp chọc hút tế bào bằng kim nhỏ (FNA). Tế bào lấy được sau đó được phân tích dưới kính hiển vi. Bảng dữ liệu[6] trong mục tài liệu tham khảo được tính từ hình ảnh dưới kính hiển vi, các thuộc tính trong bảng mô tả các đặc tính của tế bào được phân tích. Sinh viên hãy dùng thuật toán gom cụm để gom nhóm các khối u lành tính (benign) hoặc ác tính (malignant)

VI. Tài liệu tham khảo

1. Slide bài giảng lý thuyết môn Khai thác dữ liệu.
2. Han, J., Kamber, M. & Pei, J. (2012). Data mining concepts and techniques, third edition Morgan Kaufmann Publishers
3. Mall Customer Segmentation Data, https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python
4. Titanic dataset, https://data.world/nrippner/titanic-disaster-dataset
5. Recognizing hand-written digits, scikit-learn Documentation.
6. Breast Cancer Wisconsin (Diagnostic) Data Set: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)