ĐẠI HỌC SƯ PHẠM HÀ NỘI

Khoa Công Nghệ Thông Tin

BÁO CÁO BÀI TẬP LỚN

MÔN KHAI PHÁ DỮ LIỆU

Giáo viên hướng dẫn: Lê Thị Tú Kiên

Sinh viên:

1. Bùi Thị Hà – 705105025
2. Trần Vũ Phương Uyên – 705105144

HÀ NỘI, 5/2023

Lời nói đầu

Ngày nay, khi xã hội ngày càng phát triển thì lượng thông tin càng tăng lên với tốc độ bùng nổ. Lượng dữ liệu khổng lồ ấy là nguồn tài nguyên vô giá nếu như chúng ta biết cách phát hiện và khai thác những thông tin hữu ích có trong đó. Như vậy vấn đề đặt ra với dữ liệu của chúng ta là việc lưu trữ và khai thác chúng. Các phương pháp khai thác dữ liệu truyền thống ngày càng không đáp ứng được nhu cầu thực tế. Do đó, các kỹ thuật **khai phá dữ liệu** (Data mining) ra đời đã giúp giải quyết vấn đề này.

Data mining được áp dụng trong hầu hết các lĩnh vực của cuộc sống: lĩnh vực tài chính, chăm sóc sức khỏe, viễn thông, kinh doanh,… Tùy vào các bài toán khác nhau, người sử dụng sẽ sử dụng các kỹ thuật khai phá dữ liệu phù hợp. Có nhiều kỹ thuật phân tích dữ liệu thường được sử dụng trong data mining như: *Luật kết hợp, Phân lớp dữ liệu (phân lớp với cây quyết định, phân lớp Naïve Bayesian, phân lớp k phần tử gần nhất, ...), phân cụm dữ liệu (phân cụm phân hoạch (k-means, k-medoids), phân cụm phân cấp),...*

Trong phạm vi bài tập lớn này, chúng em sẽ khai phá tập dữ liệu về phân khúc khách hàng "**Mall_Customers.csv**" bằng phương pháp ***phân cụm dữ liệu* k-means và k-medoids.**

Trong quá trình làm bài tập lớn này, chúng em xin gửi lời cảm ơn đến cô **Lê Thị Tú Kiên** đã chỉ bảo và cung cấp cho chúng em những kiến thức hữu ích về Data mining, cho chúng em hiểu được hơn về tầm quan trọng của dữ liệu và làm thế nào để làm chủ những dữ liệu đó. Chúng em rất mong nhận được những lời góp ý từ cô!

Chúng xem xin chân thành cảm ơn!

Sinh viên nhóm 3

Nội dung

1 Giới thiệu

Phân khúc khách hàng là một nhiệm vụ quan trọng đối với bất kỳ doanh nghiệp nào muốn hiểu khách hàng của mình tốt hơn và cung cấp các dịch vụ được cá nhân hóa. Nó liên quan đến việc nhóm các khách hàng có đặc điểm tương tự dựa trên nhân khẩu học, hành vi và sở thích của họ, trong số các yếu tố khác. Quá trình này rất cần thiết trong việc xác định nhu cầu, sở thích và hành vi của khách hàng giúp các công ty nhắm mục tiêu và giữ chân khách hàng của họ tốt hơn. Phân khúc khách hàng hiệu quả cung cấp thông tin chi tiết về nhu cầu của các phân khúc khách hàng khác nhau và giúp phát triển các chiến lược tiếp thị mục tiêu, cải thiện sự hài lòng của khách hàng và tăng khả năng sinh lời.

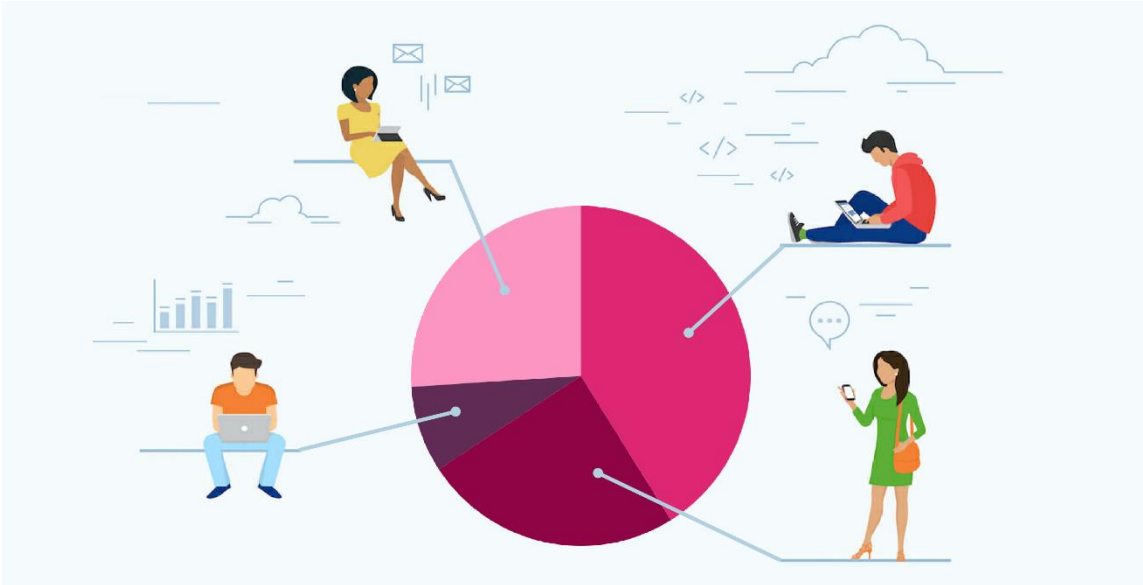


Figure 1: Hình minh hoạ cho bài toán phân khúc khách hàng

Một số phương pháp phân cụm tồn tại để phân khúc khách hàng, bao gồm phân cụm theo cấp bậc, k-means, k-medoids và phân cụm dựa trên mật độ. Các phương pháp này nhóm các khách hàng tương tự lại với nhau dựa trên các tiêu chí khác nhau, chẳng hạn như khoảng cách, tính tương đồng và mật độ. K-means và K-

medoids là các phương pháp phân cụm phổ biến được sử dụng để xác định các nhóm khách hàng riêng biệt dựa trên sự giống nhau về tính năng của chúng. K-means là phân cụm nhóm khách hàng thành k cụm, với mỗi cụm có một trọng tâm đại diện cho các giá trị trung bình của nó. Mặt khác, phân cụm K-medoids liên quan đến việc chọn k điểm dữ liệu đại diện được gọi là medoids để tạo thành các cụm.

Trong dự án này, chúng tôi sẽ sử dụng các phương pháp phân cụm K-means và K-medoids để phân khúc khách hàng dựa trên hành vi mua hàng của họ. Mục đích là để xác định các nhóm khách hàng riêng biệt có thói quen mua hàng tương tự, có thể được sử dụng để phát triển các chiến lược tiếp thị mục tiêu nhằm cải thiện sự hài lòng và giữ chân khách hàng. Bằng cách tận dụng hai phương pháp này, chúng tôi hy vọng sẽ thu được những hiểu biết có giá trị về hành vi và sở thích của khách hàng để có thể giúp các doanh nghiệp đưa ra quyết định sáng suốt nhằm cải thiện lợi nhuận của họ.

2 Mô tả bài toán

Tập dữ liệu của chúng tôi là tập *"Mall Customer"* chứa thông tin về 200 khách hàng của một trung tâm mua sắm, bao gồm 5 thuộc tính được biểu diễn trong bảng:

STT	Thuộc tính	Kiểu dữ liệu	Mô tả
1	CustomerID	Numberic	Số định danh khách hàng, mỗi khách hàng có một số định danh riêng biệt
2	Gender	Nominal	Giới tính khách hàng, có thể là Male (nam) hoặc Female (nữ)
3	Age	Numberic	Tuổi của khách hàng nằm trong khoảng từ 18 đến 70 tuổi

4	Income	Numberic	Thu nhập hàng năm của khách hàng (nghìn đô la Mỹ)
5	Spending	Numberic	Điểm số chi tiêu của khách hàng (từ 1 đến 100)

3 Các bước thực hiện

Chúng tôi thực hiện đề tài này bằng ngôn ngữ Python với một số thư viện hỗ trợ như numpy, pandas, matplotlib, sklearn, seaborn. Các bước thực hiện bao gồm đọc dữ liệu, trực quan hoá dữ liệu, xây dựng mô hình và đánh giá. Trong phần này chúng tôi sẽ trình bày chi tiết các bước này.

3.1 Đọc dữ liệu

Dữ liệu được cung cấp dưới một file “Mall_customer.csv” và được đọc vào bằng thư viện pandas:

```
df = pd.read_csv("Mall_Customers.csv")
df.isnull().sum()
```

```
CustomerID    0
Gender        0
Age           0
Income        0
Spending      0
dtype: int64
```

Có thể thấy dữ liệu không có dòng nào bị thiếu một trường nào. Tiếp theo là quan sát tổng quan về dữ liệu

```
df.describe()
```

	CustomerID	Age	Income	Spending
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Độ tuổi trung bình là 39, thấp nhất là 18 và cao nhất là 70. Tương tự với các trường khác được thể hiện trong bảng.

Tiếp tục thực hiện quan sát một số dòng đầu tiên của dữ liệu

```
df.head()
```

	CustomerID	Gender	Age	Income	Spending
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

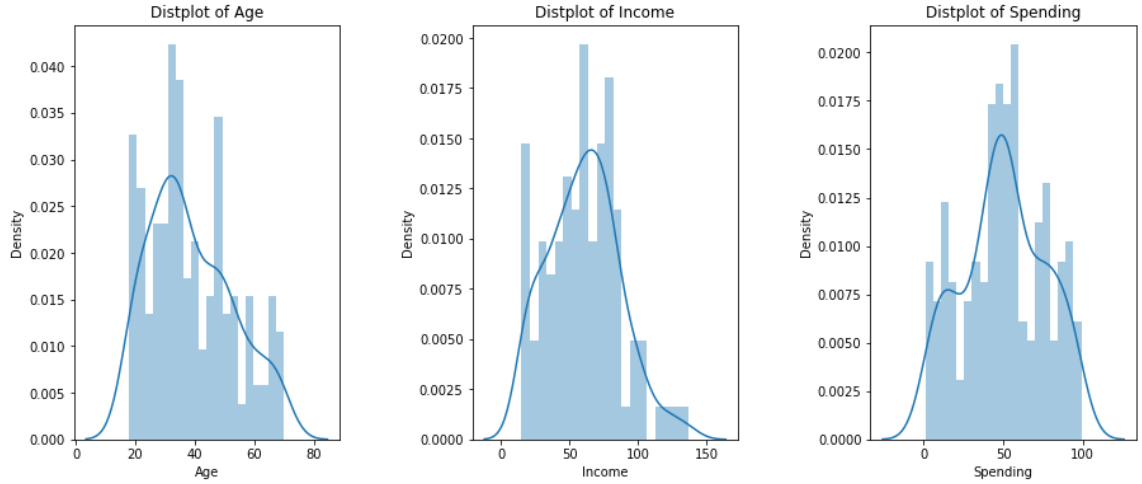
3.2 Trực quan hoá dữ liệu

Để có một cái nhìn rõ ràng hơn về dữ liệu, chúng tôi thực hiện phân tích và trực quan hoá dữ liệu bằng một số biểu đồ thông dụng.

3.2.1 Biểu đồ thống kê histogram trên từng trường

```
plt.figure(1 , figsize = (15 , 6))
n = 0
for x in ['Age' , 'Income' , 'Spending']:
    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
```

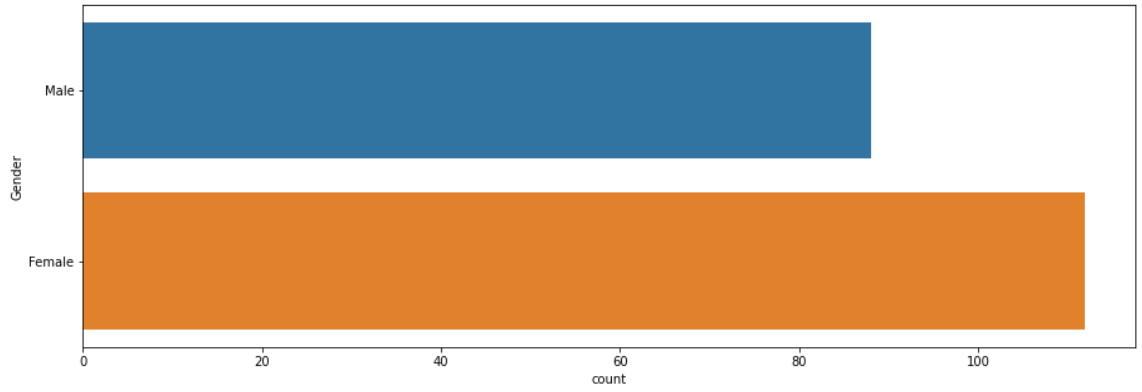
```
    sns.distplot(df[x] , bins = 20)
    plt.title('Distplot of {}'.format(x))
plt.show()
```



Mỗi biểu đồ thể hiện phân bố dữ liệu hay số khách hàng đối với từng thuộc tính (Age, Income và Spending).

3.2.2 Biểu đồ thống kê giới tính

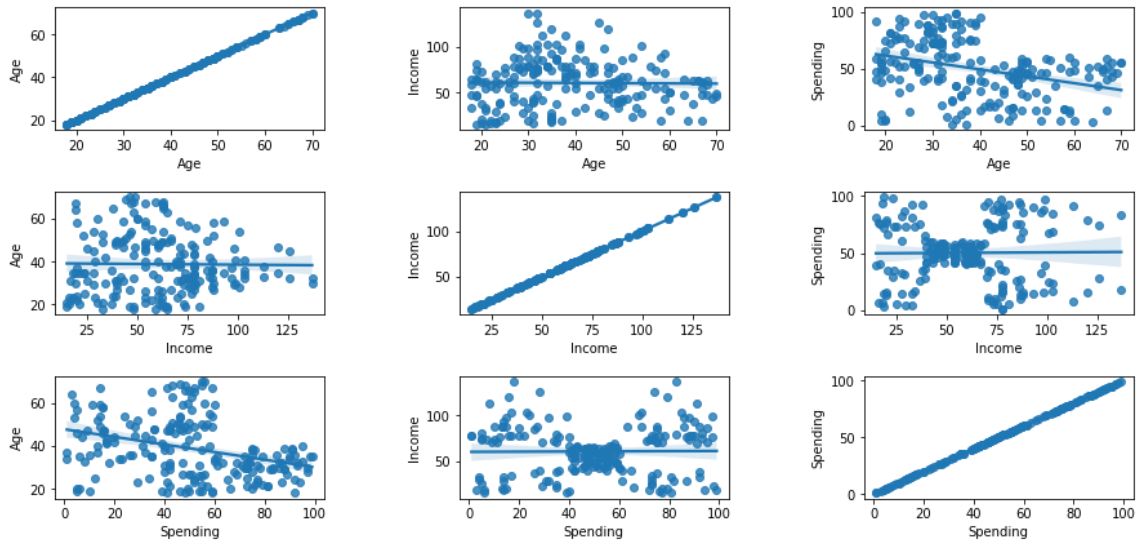
```
plt.figure(1 , figsize = (15 , 5))
sns.countplot(y = 'Gender' , data = df)
plt.show()
```



3.2.3 Biểu đồ tương quan giữa các thuộc tính

```
plt.figure(1 , figsize = (15 , 7))
n = 0
for x in ['Age' , 'Income' , 'Spending']:
    for y in ['Age' , 'Income' , 'Spending']:
        n += 1
```

```
        plt.subplot(3 , 3 , n)
        plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
        sns.regplot(x = x , y = y , data = df)
        plt.ylabel(y.split()[0]+' '+y.split()[1] if len(y.split()) > 1
else y )
plt.show()
```



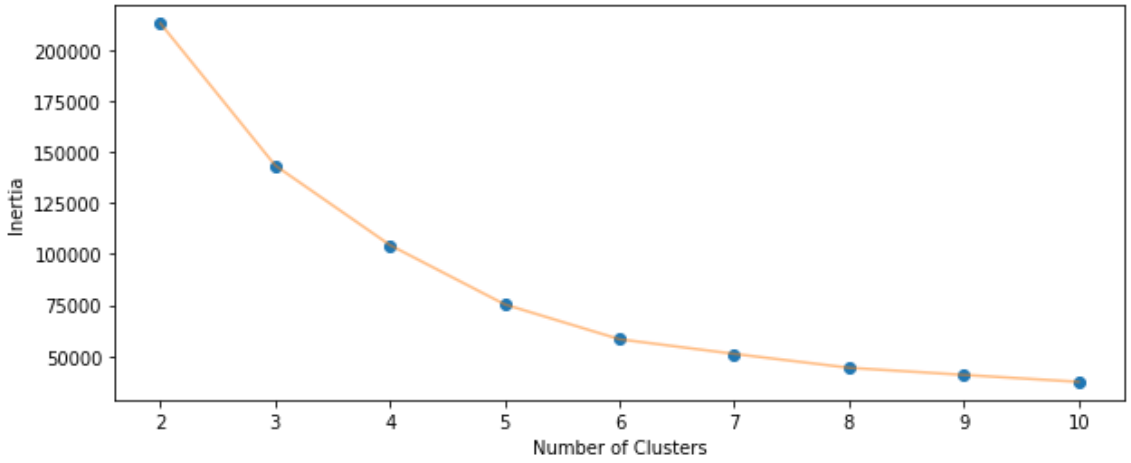
3.3 Xây dựng mô hình Kmeans

Tham số quan trọng nhất của mô hình Kmeans là số cụm. Chúng tôi tiến hành thử nghiệm với số cụm khác nhau (từ 2 đến 10). Sau đó lưu các kết quả inertia vào một mảng. Inertia của một kết quả phân cụm là tổng bình phương khoảng cách của từng điểm so với tâm cụm của nó. Sau đó chúng tôi sẽ quan sát sự biến đổi của chỉ số này để chọn ra tâm cụm phù hợp. Các thuộc tính được sử dụng bao gồm Age, Income và Spending. Chúng tôi chưa xét đến thuộc tính về giới tính trong đồ án này.

```
X = df[['Age', 'Income', 'Spending']].to_numpy()
inertia = []
for n in range(2, 11):
    algorithm = KMeans(n_clusters=n)
    algorithm.fit(X)
    inertia.append(algorithm.inertia_)
```

Vẽ đồ thị thể hiện sự biến đổi của inertia so với số cụm:

```
plt.figure(figsize=(10, 4))
plt.plot(np.arange(2 , 11) , inertia , 'o')
plt.plot(np.arange(2 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```



Số cụm tăng dần thì inertia giảm dần, điều này dễ hiểu. Để chọn số cụm tối ưu thì chúng tôi sử dụng phương pháp Elbow [2], tức là lựa chọn điểm "khuỷu tay" của đồ thị. Khi số cụm K=2 thay đổi lên K=3, lượng inertia giảm rất lớn. Tương tự như vậy cho tới khi K thay đổi từ 4 lên 5. Ở khoảng K=5 lên K=6, inertia giảm rất ít và sau đó thì gần như không thay đổi đáng kể. Vì vậy K tối ưu xác định bằng 5. Chúng tôi huấn luyện lại mô hình Kmeans với K=5:

```
X = customers[['Age', 'Income','Spending']].copy()
km = KMeans(n_clusters=5)
customers['cluster'] = km.fit_predict(X)
customers['cluster'] = customers['cluster'].astype('category')
```

Và vẽ biểu đồ trực quan cho việc phân cụm:

```
px.scatter_3d(customers,
              x='Age',
              y='Income',
              z='Spending',
              color='cluster')
```

hình 1: biểu đồ trực quan phân cụm dữ liệu mô hình k-means với k = 5

3.4 Xây dựng mô hình Kmedoids

Phần này khá tương tự với việc xây dựng mô hình Kmeans. Chỉ khác ở chỗ thay hàm tạo mô hình của từ Kmeans của sklearn thành Kmedoids của sklearn_extra.

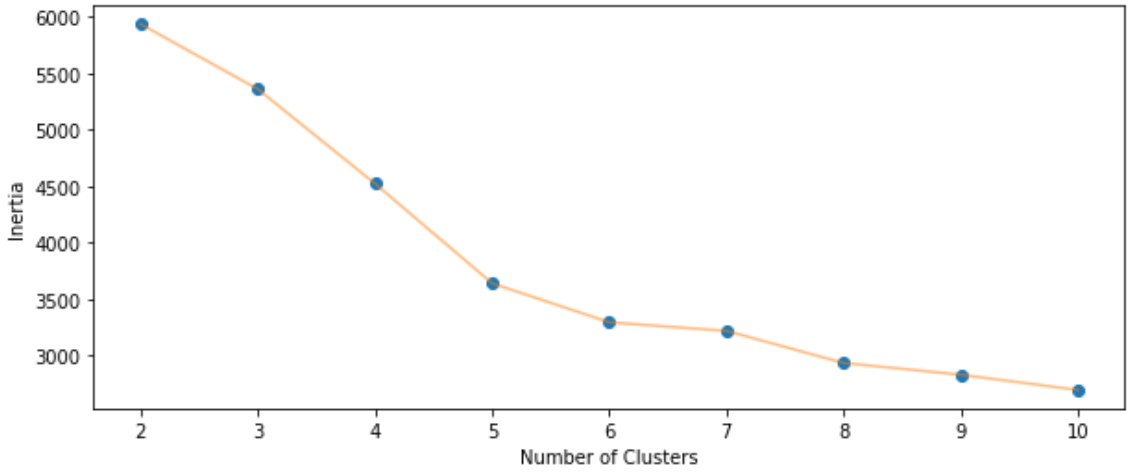
Chạy thử nghiệm với K=2 đến K=10:

```
X = df[['Age', 'Income', 'Spending']].to_numpy()
inertia = []
for n in range(2, 11):
    algorithm = KMedoids(n_clusters=n)
    algorithm.fit(X)
    inertia.append(algorithm.inertia_)
```

Vẽ biểu đồ của inertia so với K:

```
plt.figure(figsize=(10, 4))
plt.plot(np.arange(2 , 11) , inertia , 'o')
plt.plot(np.arange(2 , 11) , inertia , '-' , alpha = 0.5)
```

```
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```



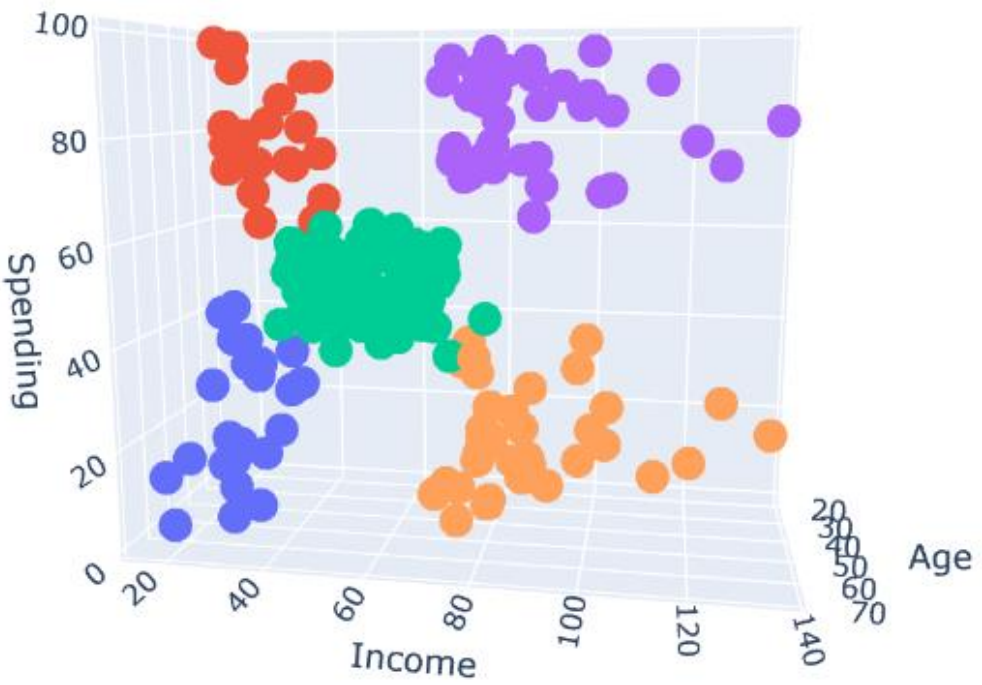
Từ biểu đồ này ta cũng thấy rõ sự thay đổi đột ngột ở điểm K=5. Vì vậy ta cũng xác định K=5 là tối ưu cho mô hình.

Huấn luyện lại mô hình với K=5:

```
X = customers[['Age', 'Income','Spending']].copy()
kmd = KMedoids(n_clusters=5)
customers['cluster'] = km.fit_predict(X)
customers['cluster'] = customers['cluster'].astype('category')
```

Kết quả trực quan:

```
px.scatter_3d(customers,
              x='Age',
              y='Income',
              z='Spending',
              color='cluster')
```



hình 2: biểu đồ trực quan phân cụm dữ liệu mô hình k-medoids với k = 5

3.5 Đánh giá kết quả

Đánh giá kết quả phân cụm giữa hai thuật toán k-means và k-medoids thông qua độ đo Silhouette

Nhìn vào hai biểu đồ kết quả phân cụm trực quan, rất khó để có thể nói rằng phương pháp nào tốt hơn phương pháp nào vì chúng rất giống nhau. Để có thể đo lường được độ hiệu quả của thuật toán phân cụm, chúng tôi sử dụng độ đo Silhouette. Độ đo này được tính thông qua trung bình khoảng cách bên trong của cụm (ký hiệu là a) và trung bình khoảng cách so với cụm gần nhất (ký hiệu là b) trên từng điểm dữ liệu. Silhouette = (b-a)/max(a,b). Sau đó có thể tính trung bình trên toàn bộ tập dữ liệu. Độ đo này có giá trị nằm trong khoảng [-1,1]. Giá trị bằng 1 là tốt nhất, bằng -1 là tệ nhất. Hình bên dưới minh hoạ cho cách tính của độ đo này:

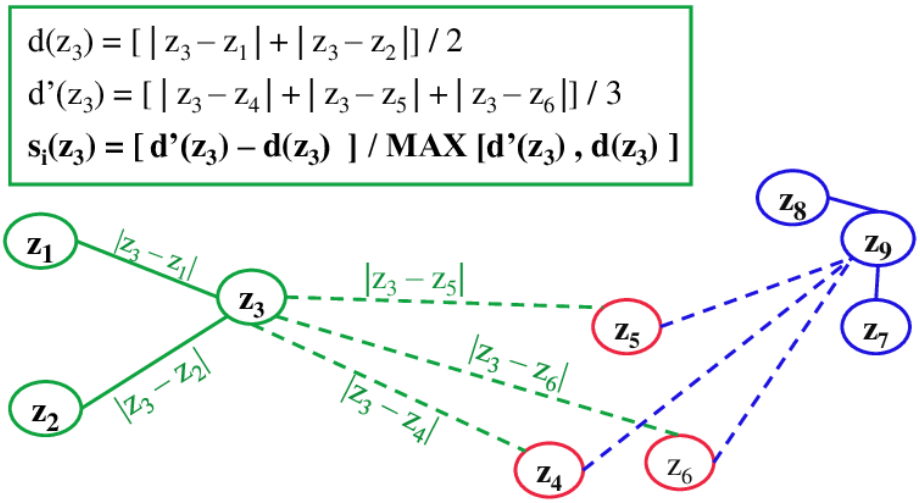


Figure 2: Minh hoạ cho cách tính độ đo Silhouette, trong đó d là khoảng cách trong cụm, d' là khoảng cách ngoài cụm gần nhất và si là độ đo Silhouette cho điểm dữ liệu thứ i.

Chúng tôi thực hiện tính toán trên kết quả của hai mô hình Kmeans và Kmedoids:

```
score = silhouette_score(X, labels, metric='euclidean')
print("Score =", score)
```

Kết quả thu được như sau:

	Kmeans	**Kmedoids**
Score	0.44	0.43

Như vậy, phương pháp Kmeans cho kết quả phân cụm tốt hơn, nhưng không đáng kể. Vậy ta có thể sử dụng cả hai phương pháp này thay thế cho nhau.

Đánh giá trực quan khi giá trị k thay đổi

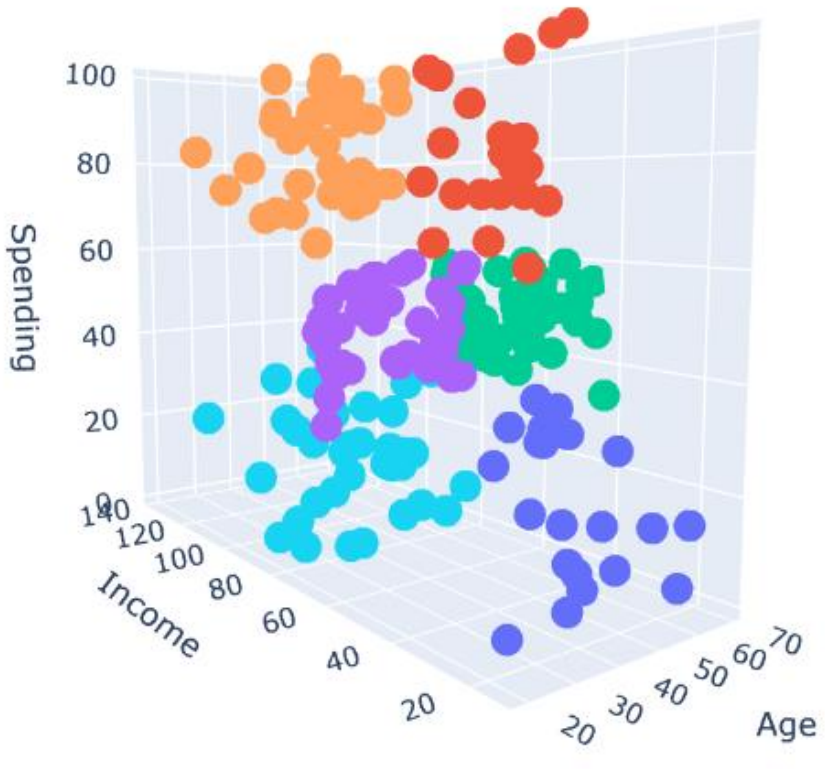
Với mô hình k-means với k = 2 ta có sơ đồ trực quan

hình 3: biểu đồ trực quan phân cụm dữ liệu mô hình k-means với k = 2

Với mô hình k-means với k = 4 ta có sơ đồ trực quan

hình 4: biểu đồ trực quan phân cụm dữ liệu mô hình k-means với k = 4

Với mô hình k-means với k = 6 ta có sơ đồ trực quan



hình 5: biểu đồ trực quan phân cụm dữ liệu mô hình k-means với k = 6

Nhìn vào các sơ đồ với các giá trị k khác nhau (k = 2, 4, 5, 6) ta thấy với giá trị k=5, ranh giới giữa các cụm được thể hiện rõ ràng nhất

4 Tính toán mô phỏng

Để hiểu rõ hơn về cách hoạt động của thuật toán, chúng tôi thực hiện tính toán mô phỏng trên một số điểm dữ liệu.

Dưới đây là kết quả phân cụm của thuật toán Kmedoids, bao gồm thông tin về điểm nào thuộc cụm nào và tâm của 5 cụm.

```
1  labels
```
✓ 0.0s

```
array([0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3,
       0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 0, 3, 4, 3, 0, 3,
       0, 3, 4, 0, 0, 0, 4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 4, 4, 4, 3,
       4, 4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
       4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 2, 1, 4, 1, 2, 1, 2, 1,
       2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
       2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
       2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1,
       2, 1], dtype=int64)
```

```
1  centroids
```
✓ 0.0s

```
array([[40., 29., 31.],
       [29., 79., 83.],
       [42., 86., 20.],
       [25., 24., 73.],
       [48., 54., 46.]])
```

Thử tính toán tâm cụm của điểm dữ liệu đầu tiên:

```
1  X[0]
```
✓ 0.0s

```
array([19, 15, 39], dtype=int64)
```

$$x_0 = (19,15,39)$$

Khoảng cách của x_0 so với 5 tâm cụm $c_0, \dots c_4$ là:

		Tính khoảng cách bằng công thức Euclide
$c_0 = (40,29,31)$	$x_0 = (19,15,39)$	26.476
$c_1 = (29,79,83)$	$x_0 = (19,15,39)$	78.307

$c_2 = (42,86,20)$	$x_0 = (19,15,39)$	77.013
$c_3 = (25,24,73)$	$x_0 = (19,15,39)$	35.679
$c_4 = (48,54,46)$	$x_0 = (19,15,39)$	49.102

Vậy khoảng cách từ x_0 đến $c_0$là nhỏ nhất, chứng tỏ $x_0$thuộc cụm số 0. Kết quả này hoàn toàn trùng khớp với kết quả trong mảng labels.

Tương tự với 9 điểm tiếp theo, chúng tôi có bảng kết quả tính toán sau đây:

$d(x_i, c_j)$	c_0	c_1	c_2	c_3	c_4	Cụm
x_0	26.476	78.307	77.013	35.679	49.102	0
x_1	55.29	64.529	95.932	12.689	58.949	3
x_2	34.554	99.895	74.699	67.661	61.871	0
x_3	50.735	63.569	92.25	9.165	55.045	3
x_4	17.493	75.478	72.677	34.264	41.158	0
x_5	49.93	62.785	91.088	8.185	54.268	3
x_6	27.767	98.417	69.778	68.007	55.362	0
x_7	66.174	62.274	102.279	21.932	65	3
x_8	38.21	105.948	72.54	80.287	57.706	0
x_9	43.37	61.008	85.656	7.141	47.17	3

Kết quả này hoàn toàn trùng khớp với nhãn của mô hình Kmedoids đã được tính.

5 Kết luận

Trong bài tập lớn này, chúng tôi đã tiến hành tìm hiểu và xây dựng mô hình Kmeans, Kmedoids để phục vụ cho bài toán phân khúc khách hàng. Kết quả tính toán cho thấy cả hai mô hình đều có hiệu năng tốt như nhau trên tập dữ liệu Mall_Customers.csv với số cụm tối ưu được xác định bằng 5 và độ đo Silhouette khoảng 0.45.

Qua bài tập lớn này, chúng tôi đã được trau đồi kiến thức về học máy nói chung và bài toán phân cụm nói riêng. Chúng tôi đã hiểu rõ cách hoạt động và vận dụng thành công hai phương pháp phân cụm kinh điển là Kmeans và Kmedoids cho bài toán của mình.

6 Tài liệu tham khảo

[1] V. Choudhary, “Mall Customer Segmentation Data,” Kaggle, https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-tutorial-in-python (accessed May 3, 2023).

[2] Robert L. Thorndike (December 1953). "Who Belongs in the Family?". Psychometrika. 18 (4): 267–276. doi:10.1007/BF02289263. S2CID 120467216.