



Báo cáo đồ án Customer Personality Analysis



Môn học: Dữ liệu lớn - IS405.O21.HTCL

GVHD: ThS . Nguyễn Hồ Duy Tri

Sinh viên thực hiện:

- Lý Gia Hiếu - 21522074
- Thiều Vĩnh Tiến - 21521533
- Man Ngô Thủy Tiên - 21521526
- Trần Tịnh Minh Tú - 21521619



Nội dung

1

Giới thiệu tổng quan

2

Tiền xử lý dữ liệu

3

Triển khai thuật toán

4

Kết quả đạt được và
Kết luận



I. Giới thiệu tổng quan





1. Lý do chọn đề tài

01

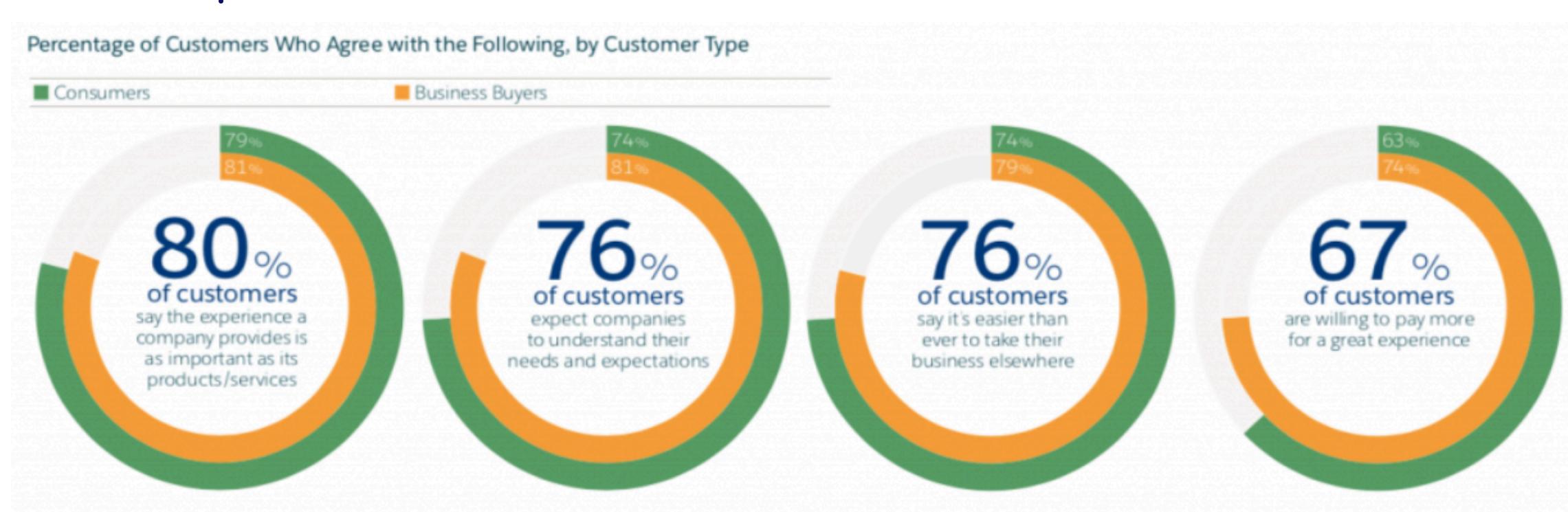
Phân tích chân dung khách hàng là một khía cạnh quan trọng trong **kinh doanh** và **marketing** hiện đại

02

Giúp doanh nghiệp tiếp cận đối tượng mục tiêu chính xác, **tối ưu hóa các hoạt động kinh doanh** (phát triển sản phẩm đến chiến lược marketing), tạo trải nghiệm cá nhân hóa cho KH, tăng cường mối quan hệ giữa KH và thương hiệu.

03

Theo Salesforce: Khảo sát hơn 6000 người tiêu dùng: 76% mong đợi các công ty hiểu được **nhu cầu** của họ





1. Lý do chọn đề tài

Với những lý do trên nhóm lựa chọn đề tài **Phân cụm chân dung khách hàng qua thuật toán phân cụm**. Phương pháp này sẽ giúp tổ chức và phân loại khách hàng thành các nhóm dựa trên đặc điểm chung từ đó các doanh nghiệp có thể phát triển các chiến lược đích thực hướng đến từ nhóm khách hàng một cách tối ưu, phù hợp với ngành học TMĐT của nhóm.



2.1 Nguồn dữ liệu



2. Mô tả dữ liệu

- **Tên bộ dữ liệu:** Customer Personality Analysis
- **Tác giả:** Dr. Omar Romero-Hernandez
- Bộ dữ liệu có sẵn trên Kaggle tại đường dẫn: Customer Personality Analysis
- Dữ liệu bao gồm **29 thuộc tính x 2240 dòng** chứa thông tin về các khách hàng và các thuộc tính liên quan đến hành vi mua sắm và thói quen tiêu dùng của họ.



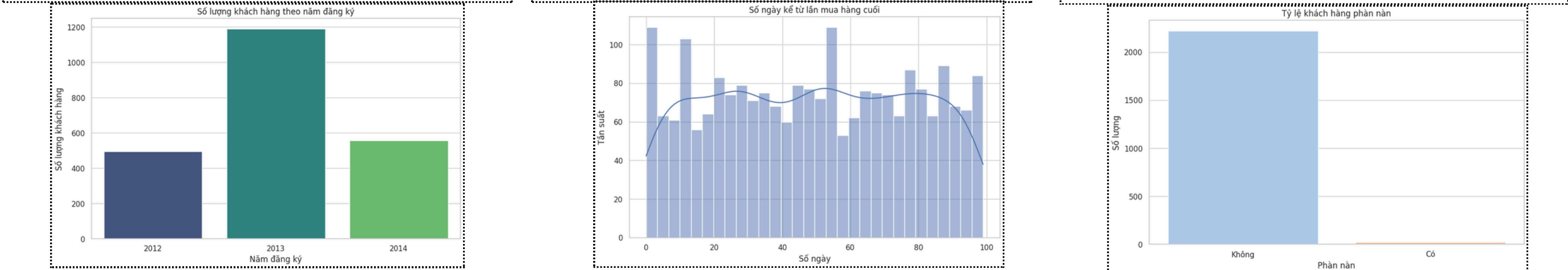
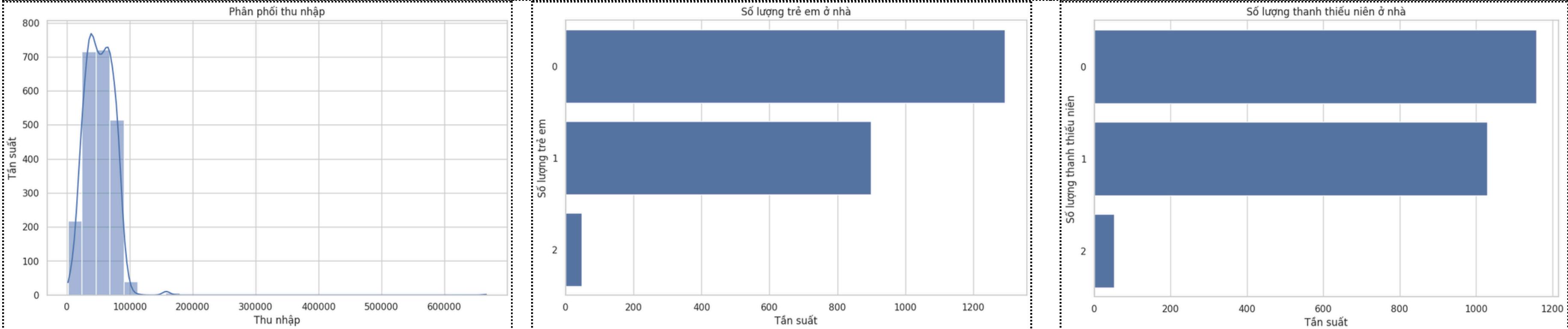
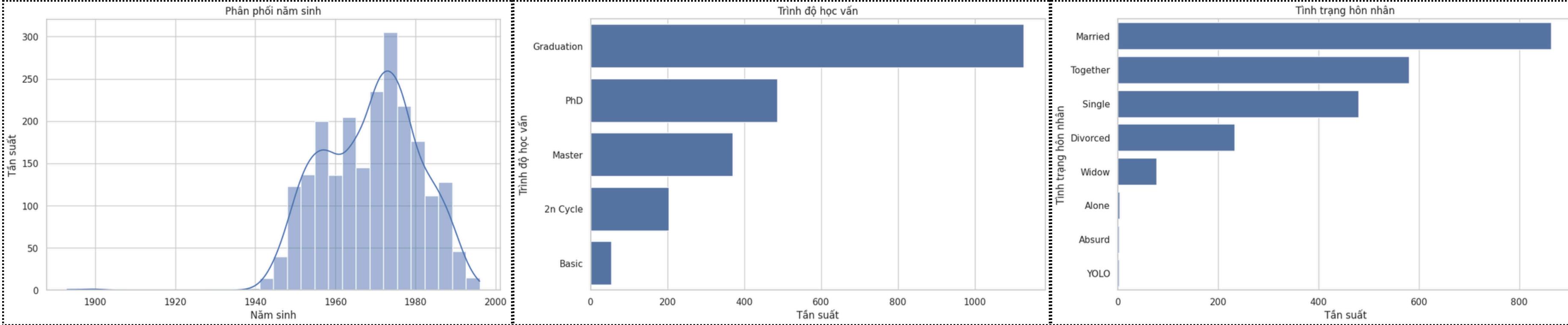
2. Mô tả dữ liệu



People

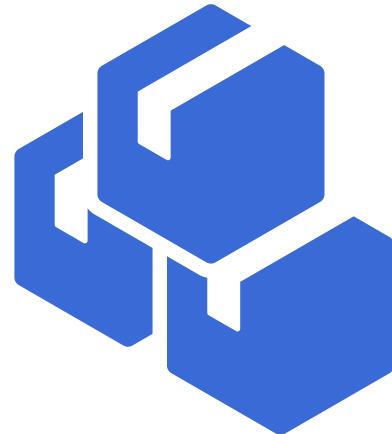
2.2 Mô tả đặc trưng

STT	Tên thuộc tính	Kiểu dữ liệu	Nội dung
1	ID	int	Mã định danh duy nhất của khách hàng
2	Year_Birth	int	Năm sinh của khách hàng
3	Education	string	Trình độ học vấn của khách hàng
4	Marital_Status	string	Tình trạng hôn nhân của khách hàng
5	Income	int	Thu nhập hộ gia đình hàng năm của khách hàng
6	Kidhome	int	Số trẻ em trong gia đình khách hàng
7	Teenhome	int	Số thanh thiếu niên trong gia đình khách hàng
8	Dt-Customer	string	Ngày khách hàng đăng ký với công ty
9	Recency	int	Số ngày kể từ lần mua hàng cuối cùng của khách hàng
10	Complain	int	1 nếu khách hàng phàn nàn trong 2 năm qua, 0 nếu ngược lại





2. Mô tả dữ liệu

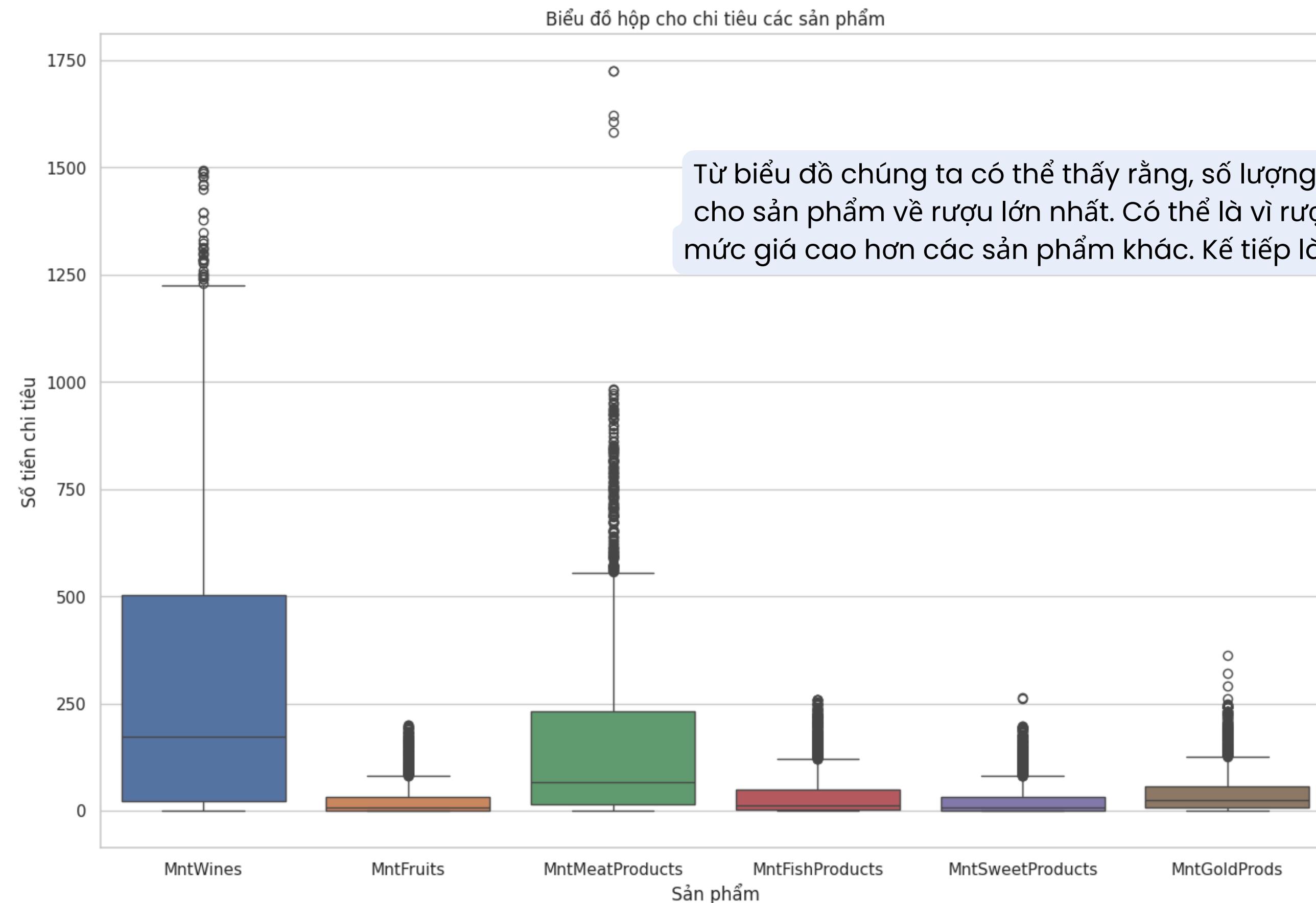


Product

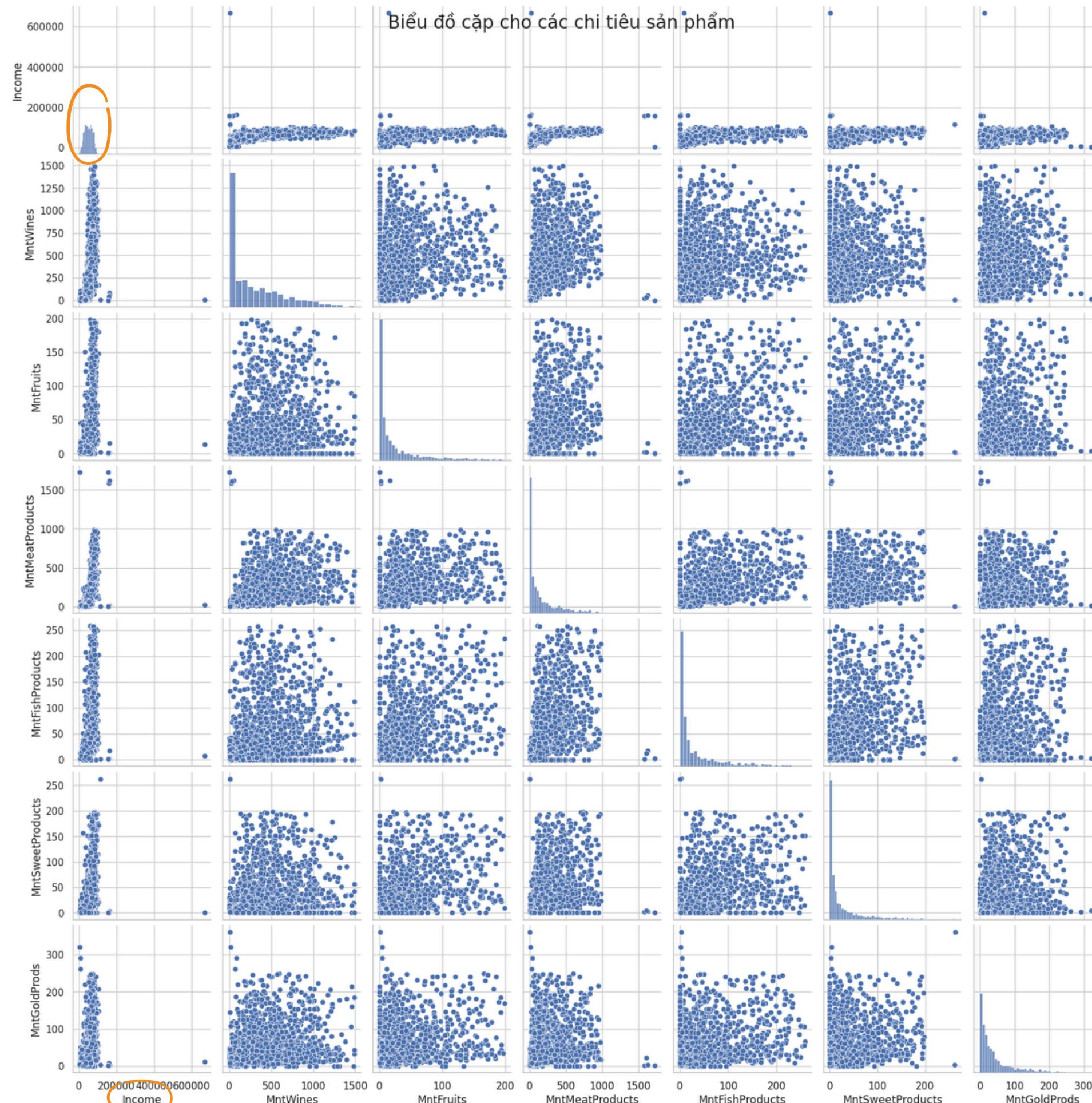
2.2 Mô tả đặc trưng

STT	Tên thuộc tính	Kiểu dữ liệu	Nội dung
11	MntWines	int	Số tiền chi cho rượu vang trong 2 năm qua
12	MntFruits	int	Số tiền chi cho trái cây trong 2 năm qua
13	MntMeatProducts	int	Số tiền chi cho thịt trong 2 năm qua
14	MntFishProducts	int	Số tiền chi cho cá trong 2 năm qua
15	MntSweetProducts	int	Số tiền chi cho đồ ngọt trong 2 năm qua
16	MntGoldPods	int	Số tiền chi cho vàng trong 2 năm qua

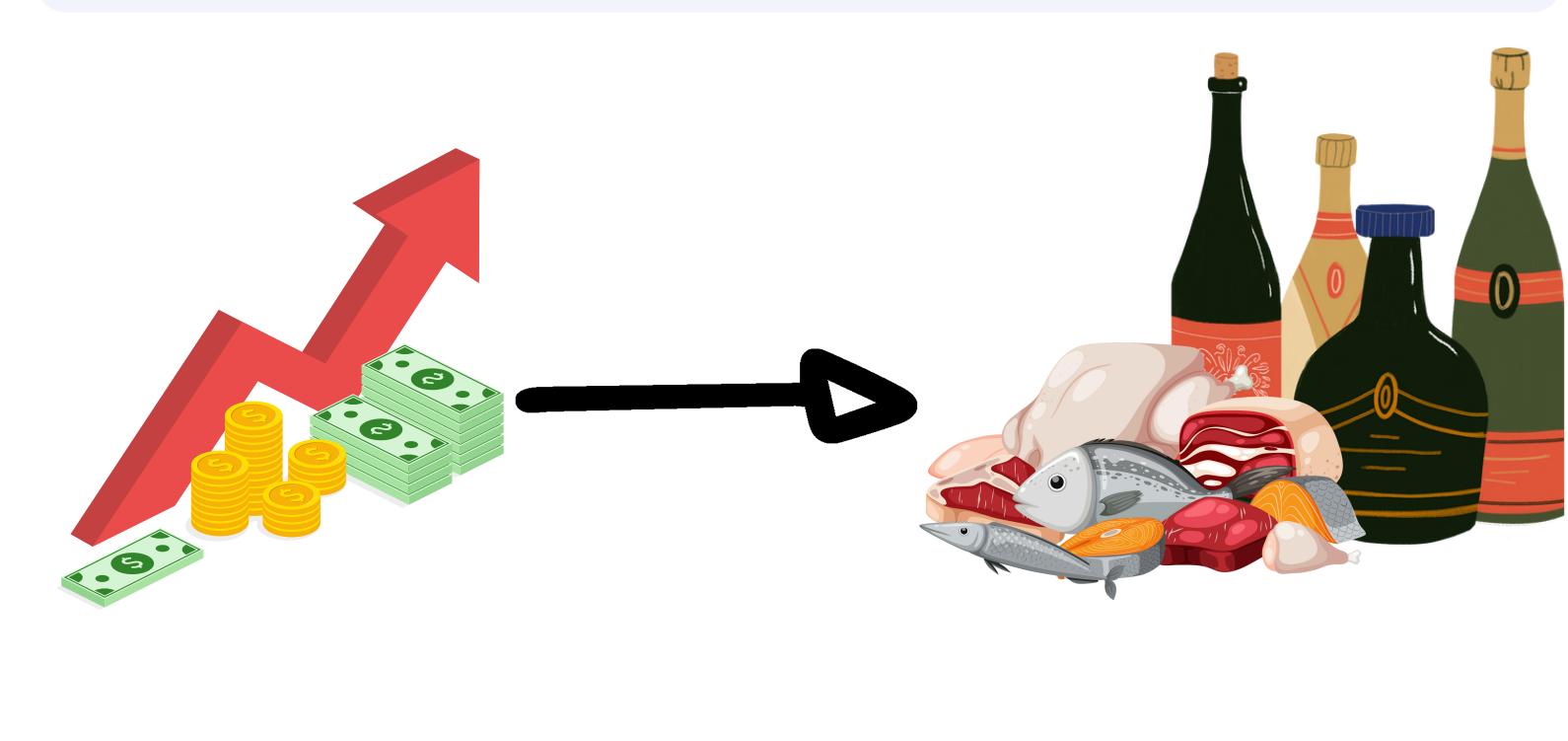
Khám phá mối quan hệ giữa Số tiền chi tiêu và sản phẩm



Khám phá mối quan hệ giữa thu nhập và số lượng chi tiêu cho các sản phẩm

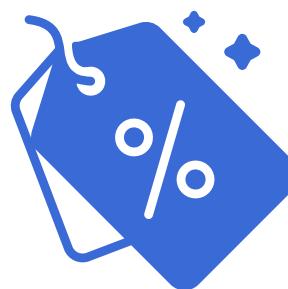


- Biểu đồ histogram cho Income cho thấy phần lớn các cá nhân có **thu nhập tập trung** ở một phạm vi nhất định.
- Mối quan hệ giữa **Income và MntWines** có thể quan sát được **xu hướng dương** có thể cho thấy thu nhập cao thường đi kèm với chi tiêu cho rượu vang cao hơn.
- Tương tự, thu nhập cao hơn có thể liên quan đến chi tiêu cao hơn cho sản phẩm thịt và cá





2. Mô tả dữ liệu



Promotion

2.2 Mô tả đặc trưng

STT	Tên thuộc tính	Kiểu dữ liệu	Nội dung
17	NumDealsPurchases	int	Số lần mua hàng được giảm giá
18	AcceptedCmp1	int	1 nếu khách hàng chấp nhận ưu đãi trong chiến dịch đầu tiên, 0 nếu không
19	AcceptedCmp2	int	1 nếu khách hàng chấp nhận ưu đãi trong chiến dịch thứ 2, 0 nếu ngược lại
20	AcceptedCmp3	int	1 nếu khách hàng chấp nhận ưu đãi trong chiến dịch thứ 3, 0 nếu ngược lại
21	AcceptedCmp4	int	1 nếu khách hàng chấp nhận ưu đãi trong chiến dịch thứ 4, 0 nếu ngược lại
22	AcceptedCmp5	int	1 nếu khách hàng chấp nhận ưu đãi trong chiến dịch thứ 5, 0 nếu ngược lại
23	Response	int	1 nếu khách hàng chấp nhận ưu đãi trong chiến dịch trước, 0 nếu không



2. Mô tả dữ liệu

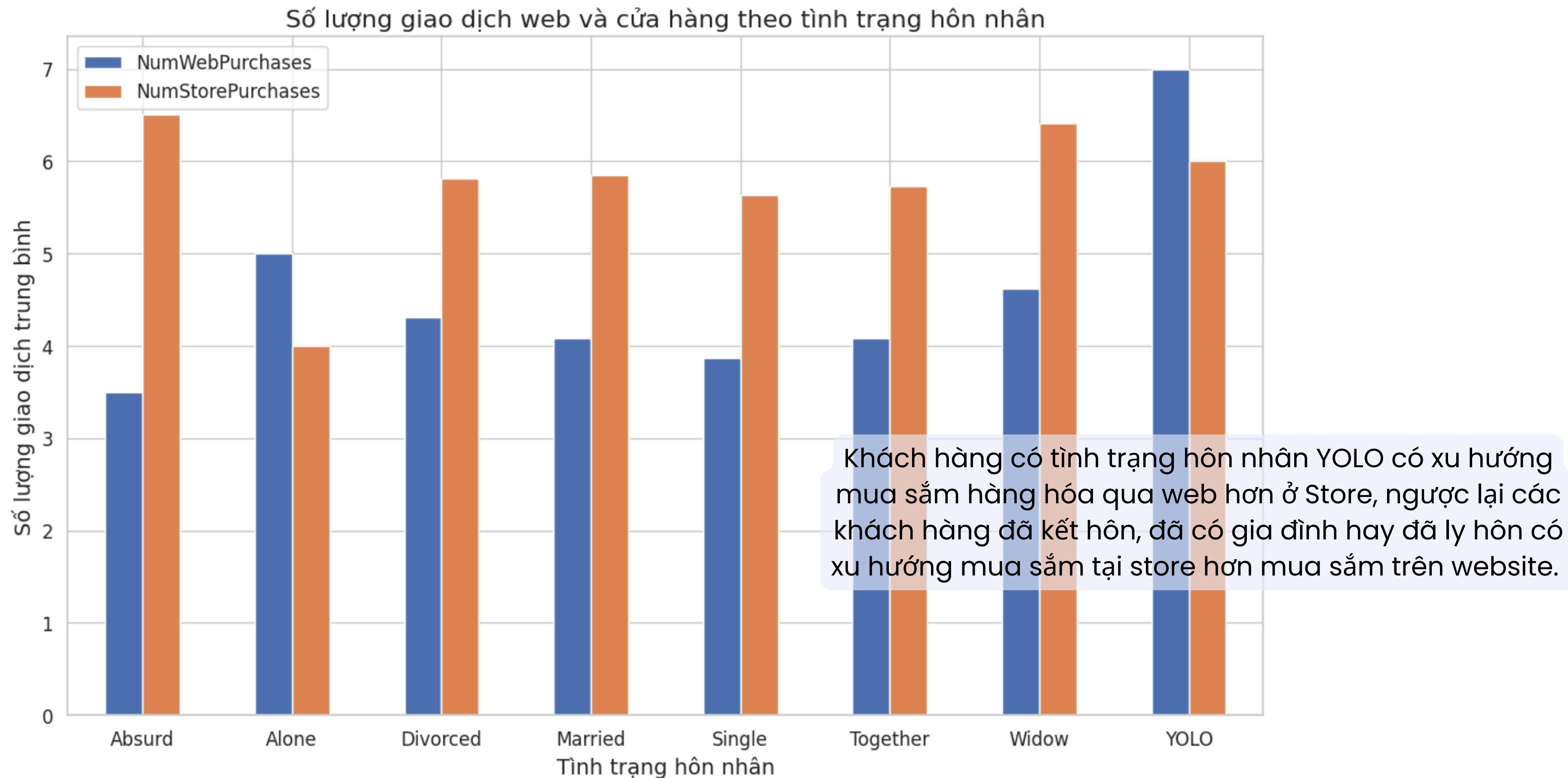


Place

2.2 Mô tả đặc trưng

STT	Tên thuộc tính	Kiểu dữ liệu	Nội dung
24	NumWebPurchases	int	Số lượng mua hàng được thực hiện thông qua trang web của công ty
25	NumCatalogPurchases	int	Số lần mua hàng được thực hiện bằng danh mục
26	NumStorePurchases	int	Số lượng mua hàng được thực hiện trực tiếp tại cửa hàng
27	NumWebVisitsMonth	int	Số lượt truy cập vào trang web của công ty trong tháng trước

Khám phá mối quan hệ giữa tình trạng hôn nhân và số lượt giao dịch web





3. Mô tả bài toán

Nhóm thực hiện các phương pháp phân cụm bằng thuật toán **KMeans** và **Hierarchical Clustering** dựa trên sự tương đồng của dữ liệu và **tìm ra các cụm khách hàng có cùng đặc điểm**.

Từ đó giúp doanh nghiệp có thể tập trung vào **phân khúc khách hàng** của họ để thực hiện các chiến lược kinh doanh phù hợp.

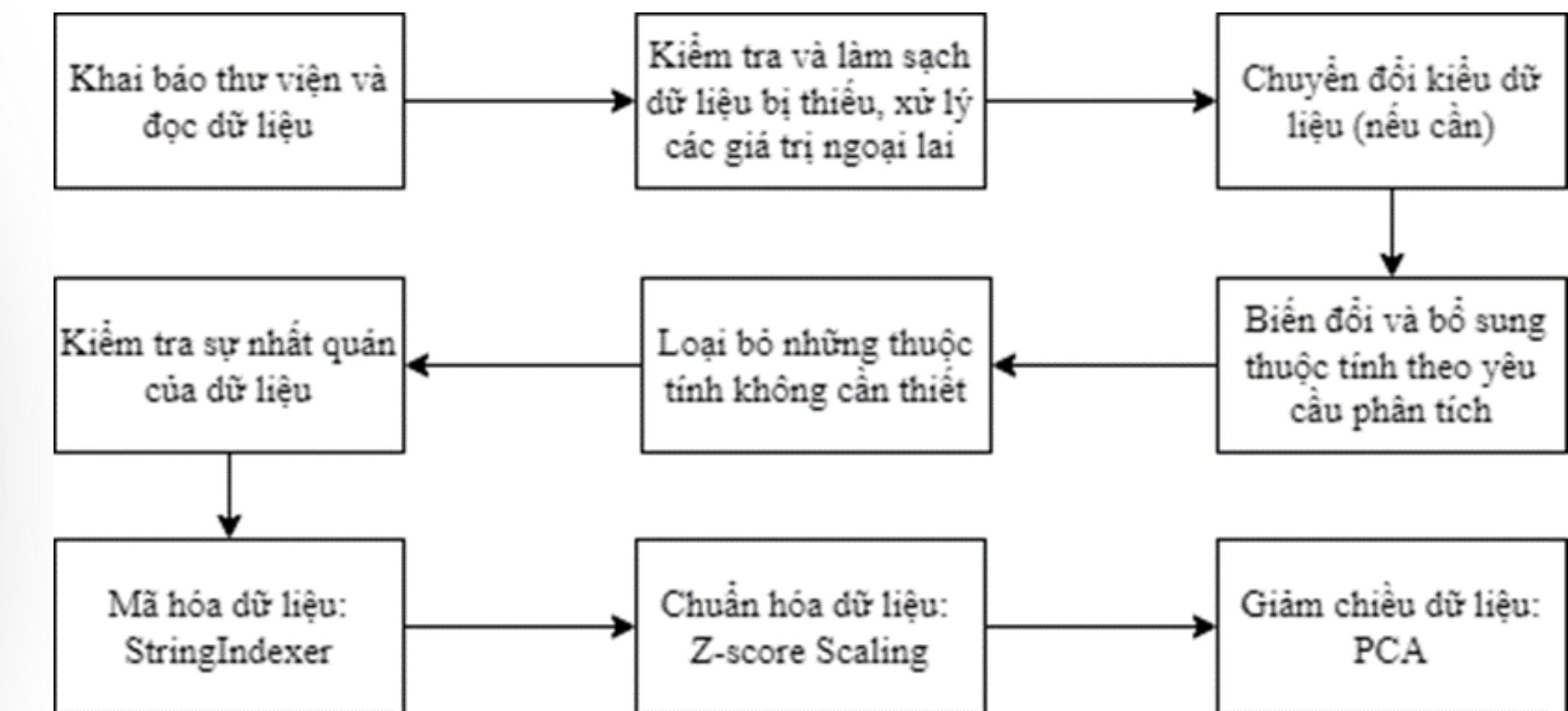


II. Tiền xử lý dữ liệu





1. Các kỹ thuật tiền xử lý dữ liệu





B1. Khai báo TV và đọc dữ liệu

```
[ ] # Importing the PySpark Libraries
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *

# Start Spark session
spark = SparkSession.builder \
    .appName("DataPreprocessing") \
    .getOrCreate()

# Importing other necessary libraries
import numpy as np
import pandas as pd
import datetime
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import colors
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import PCA
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.colors import ListedColormap
from sklearn import metrics
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
np.random.seed(42)

# PySpark MLlib for machine learning
from pyspark.ml.feature import StringIndexer, StandardScaler as SparkStandardScaler, PCA as SparkPCA
from pyspark.ml.clustering import KMeans as SparkKMeans
from pyspark.ml.evaluation import ClusteringEvaluator

# Note: Seaborn and Yellowbrick are Python-specific libraries for visualization and will not work with PySpark directly.
# You will need to convert PySpark DataFrames to Pandas DataFrames for visualization purposes when necessary.
```

Khai báo thư viện đồng thời bắt đầu một Spark Session



Import dữ liệu và đọc dữ liệu **marketing_campaign.csv** bằng spark.read.csv



```
[ ] # Define the file path  
#file_path = "/content/drive/MyDrive/NĂM 3/HK2/BigData/Dataset/marketing_campaign.csv"  
file_path = "/content/marketing_campaign.csv"  
  
# Read the dataset  
data = spark.read.csv(file_path, sep="\t", header=True, inferSchema=True)
```

B2. Kiểm tra và xử lý Null, Outliers

Xem dữ liệu

```
[ ] # Show the number of datapoints  
num_data_points = data.count()  
print("Number of datapoints:", num_data_points)  
  
# Display the first few rows of the dataset  
data.show(5)
```

ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts	MntSweetProducts	MntGoldProds	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	GlobalScore
5524	1957	Graduation	Single	58138	0	0	04-09-2012	58	635	88	546	172	88	88	3	8	10		
2174	1954	Graduation	Single	46344	1	1	08-03-2014	38	11	1	6	2	1	6	2	1	1		
4141	1965	Graduation	Together	71613	0	0	21-08-2013	26	426	49	127	111	21	42	1	8	2		
6182	1984	Graduation	Together	26646	1	0	10-02-2014	26	11	4	20	10	3	5	2	2	0		
5324	1981	PhD	Married	58293	1	0	19-01-2014	94	173	43	118	46	27	15	5	5	3		

Dữ liệu bao gồm 29 thuộc tính x 2240 dòng



Khai báo hàm **Col, Count, Isnan, When** của thư viện `pyspark.sql.function` để đếm số dữ liệu không **Null** ở từng cột. Đồng thời xem lại các **kiểu dữ liệu** của từng cột bằng `printSchema()` đã hợp lý chưa.

```
[ ] # Get information about the features
# Print the schema of the DataFrame
data.printSchema()

# Describe the DataFrame to get summary statistics
data.describe().show()

# Count non-null entries for each column to mimic pandas' info() method
from pyspark.sql.functions import col, count, isnan, when

def count_not_null(c):
    return count(when(col(c).isNotNull() & ~isnan(col(c))), c)

non_null_counts = data.agg(*[count_not_null(c).alias(c) for c in data.columns])
non_null_counts.show()
```

summary	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer
count	2240	2240	2240	2240	2216	2240	2240	224
mean	5592.159821428571	1968.8058035714287	NULL	NULL	52247.25135379061	0.44419642857142855	0.50625	NULL
stddev	3246.6621975643416	11.984069456885827	NULL	NULL	25173.076660901414	0.5383980977345935	0.5445382307698761	NULL
min	0	1893	2n Cycle	Absurd	1730	0	0	01-01-201
max	11191	1996	PhD	YOLO	666666	2	2	31-12-201

Nhận xét:

- Ta thấy ở cột “Income” có các missing value (null) vì count= 2216
- Cách xử lý: Bỏ các hàng có giá trị null



B2. Kiểm tra và xử lý Null, Outliers

```
[ ] # Remove NA values  
data = data.na.drop()  
  
# Print the total number of data points after removing rows with missing values  
print("The total number of data points after removing the rows with missing values is:", data.count())
```

⇒ The total number of data points after removing the rows with missing values is: 2216

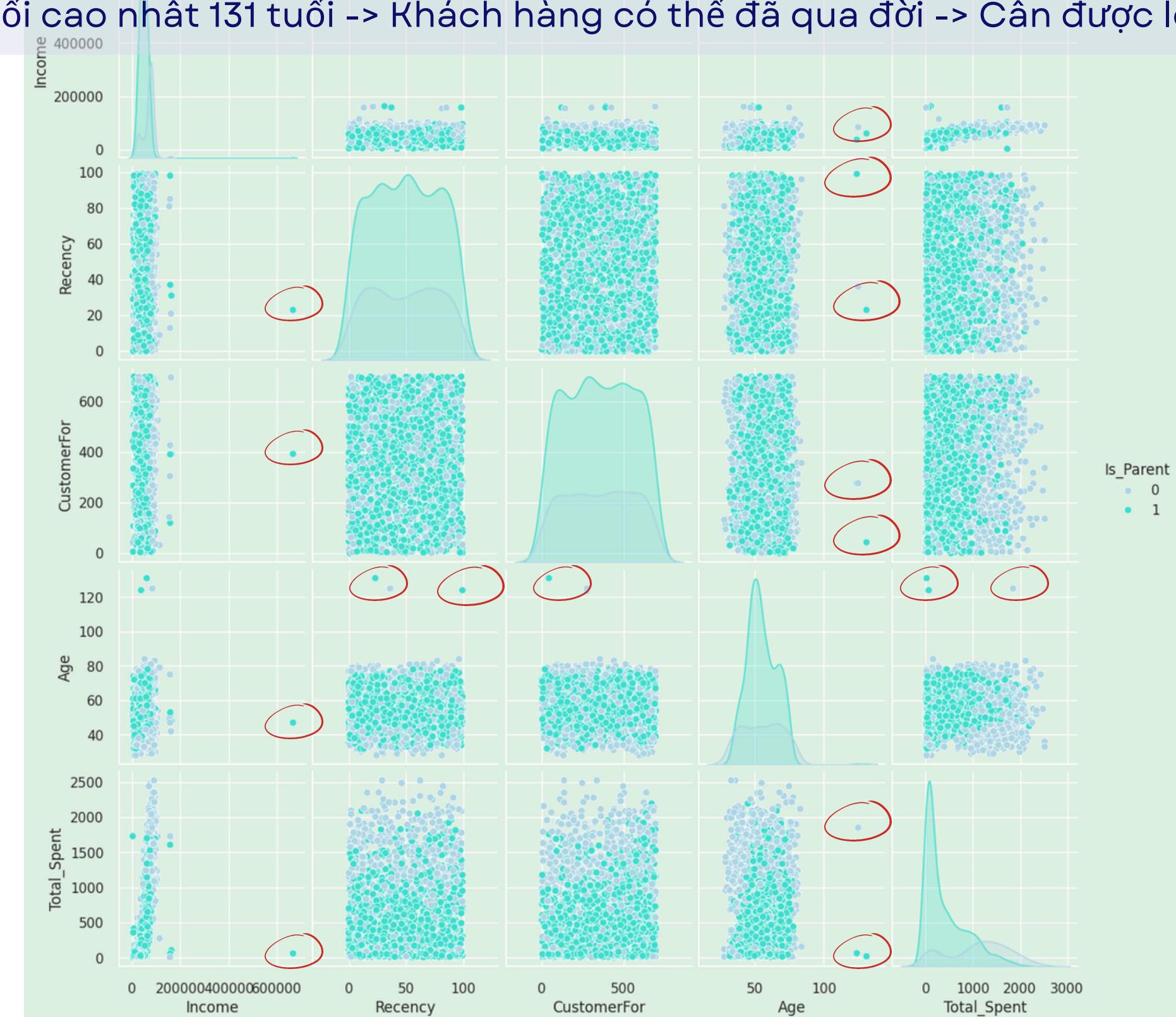
Kết quả: **Tổng số lượng điểm dữ liệu sau khi loại bỏ các dữ liệu trống: 2216**



B2. Kiểm tra và xử lý Null, Outliers

Phát hiện outlier nằm ở Age và Income. Xác định được điều kiện để loại bỏ outlier là **Age <90** và **Income <600000**

Sử dụng Scatter Plot để xem sự phân bố một số thuộc tính liên quan có xảy ra sự vô lí. Ví dụ: Độ tuổi cao nhất 131 tuổi -> Khách hàng có thể đã qua đời -> Cần được loại bỏ





Ta còn phát hiện ở **Dt_Customer** là kiểu dữ liệu **string**. Nhưng đây là **ngày** khách hàng đăng ký với công ty. Vì thế ta sẽ thay đổi kiểu dữ liệu từ **string** sang **timestamp**.



B3. Chuyển đổi kiểu dữ liệu (Dt_Customer)

```
date_format = "dd-MM-yyyy"
data = data.withColumn("Dt_Customer", to_timestamp(data["Dt_Customer"], date_format))
# Print schema to verify data types
data.printSchema()

# Show the first few rows of the cleaned data
data.show()
```

ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	Mnt
5524	1957	Graduation	Single	58138	0	0	2012-09-04 00:00:00	58	635	88	546	
2174	1954	Graduation	Single	46344	1	1	2014-03-08 00:00:00	38	11	1	6	
4141	1965	Graduation	Together	71613	0	0	2013-08-21 00:00:00	26	426	49	127	
6182	1984	Graduation	Together	26646	1	0	2014-02-10 00:00:00	26	11	4	20	
5324	1981	PhD	Married	58293	1	0	2014-01-19 00:00:00	94	173	43	118	
7446	1967	Master	Together	62513	0	1	2013-09-09 00:00:00	16	520	42	98	
965	1971	Graduation	Divorced	55635	0	1	2012-11-13 00:00:00	34	235	65	164	
6177	1985	PhD	Married	33454	1	0	2013-05-08 00:00:00	32	76	10	56	
4855	1974	PhD	Together	30351	1	0	2013-06-06 00:00:00	19	14	0	24	
5899	1950	PhD	Together	5648	1	1	2014-03-13 00:00:00	68	28	0	6	
387	1976	Basic	Married	7500	0	0	2012-11-13 00:00:00	59	6	16	11	
2125	1959	Graduation	Divorced	63033	0	0	2013-11-15 00:00:00	82	194	61	480	
8180	1952	Master	Divorced	59354	1	1	2013-11-15 00:00:00	53	233	2	53	
2569	1987	Graduation	Married	17323	0	0	2012-10-10 00:00:00	38	3	14	17	
2114	1946	PhD	Single	82800	0	0	2012-11-24 00:00:00	23	1006	22	115	
9736	1980	Graduation	Married	41850	1	1	2012-12-24 00:00:00	51	53	5	19	
4939	1946	Graduation	Together	37760	0	0	2012-08-31 00:00:00	20	84	5	38	
6565	1949	Master	Married	76995	0	1	2013-03-28 00:00:00	91	1012	80	498	
2278	1985	2n Cycle	Single	33812	1	0	2012-11-03 00:00:00	86	4	17	19	
9360	1982	Graduation	Married	37040	0	0	2012-08-08 00:00:00	41	86	2	73	

only showing top 20 rows



Tìm số ngày mà khách hàng đã gắn bó với cửa hàng (Customer For)

```
[ ] from pyspark.sql import functions as F  
  
max_date=data.select(F.max('Dt_customer').alias('max_date')).show()
```

```
+-----+  
|      max_date|  
+-----+  
|2014-06-29 00:00:00|  
+-----+
```

```
[ ] from pyspark.sql import functions as F  
data.select(F.min('Dt_customer').alias('min_date')).show()
```

```
+-----+  
|      min_date|  
+-----+  
|2012-07-30 00:00:00|  
+-----+
```

```
[ ] d1 = data.agg(max("Dt_Customer")).collect()[0][0]  
print(d1)
```

```
2014-06-29 00:00:00
```

```
[ ] d1 = data.agg(max("Dt_Customer")).collect()[0][0]  
data = data.withColumn("CustomerFor", datediff(lit(d1),col("Dt_Customer")))  
data = data.withColumn("CustomerFor", col("CustomerFor").cast(IntegerType()))
```

B4. Biến đổi và bổ sung các thuộc tính cần thiết



Để thuận tiện hơn cho việc phân loại. Chúng ta sẽ thay đổi 1 số biến cũng như loại bỏ những biến không cần thiết ra khỏi dataset



B4. Biến đổi và bổ sung các thuộc tính cần thiết

```
[ ] # Thay vì để ngày sinh chúng ta sẽ tạo ra 1 biến tuổi của khách hàng cho đến hiện tại.  
data = data.withColumn("Age", 2024 - col("Year_Birth"))  
# Tính tổng giá trị mà 1 khách hàng đã mua ở cửa hàng từ lúc tham gia  
data = data.withColumn("Total_Spent", col("MntWines") + col("MntFruits") + col("MntMeatProducts") + col("MntFishProducts") + col("MntSweetProducts"))  
# Gộp 2 cột có con nhỏ tuổi và con lớn vào thành có con  
data = data.withColumn("Children", col("Kidhome") + col("Teenhome"))  
# Ở Cột Marital_Status mặc dù nó chia ra thành độc thân hay kết hôn hay sống chung  
# Vì thế ta sẽ tạo 1 cột thay thế cột này với 2 segment là 1 mình hoặc sống theo cặp  
data = data.withColumn("Living_With",  
    when((col("Marital_Status") == "Married") | (col("Marital_Status") == "Together"), "Partner")  
    .when((col("Marital_Status") == "Widow") | (col("Marital_Status") == "YOLO") | (col("Marital_Status") == "Divorced") | (col("Marital_Status") == "Absurd"), "Alone")  
    .otherwise(col("Marital_Status")))  
# Tương tự như thế chúng ta trình độ học vấn thành 3 segment: Undergraduate, Graduation và Master  
data = data.withColumn("Education",  
    when(col("Education").isin(["Basic", "2n Cycle"]), "Undergraduate")  
    .when(col("Education").isin(["Graduation"]), "Graduate")  
    .when(col("Education").isin(["Master", "PhD"]), "Postgraduate")  
    .otherwise(col("Education")))  
# Đổi tên column sản phẩm  
data = data.withColumnRenamed("MntWines", "Wines") \  
    .withColumnRenamed("MntFruits", "Fruits") \  
    .withColumnRenamed("MntMeatProducts", "Meat") \  
    .withColumnRenamed("MntFishProducts", "Fish") \  
    .withColumnRenamed("MntSweetProducts", "Sweets") \  
    .withColumnRenamed("MntGoldProds", "Gold")  
  
[ ] # Sau khi đã biến đổi dữ liệu, bây giờ ta sẽ loại bỏ những cột ko cần thiết  
# Xác định các cột sẽ loại bỏ  
to_drop = ["Marital_Status", "Dt_Customer", "Z_CostContact", "Z_Revenue", "Year_Birth", "ID"]  
data = data.drop(*to_drop)  
data.show()  
  
[ ] #Bổ sung cột Is_Parent vì khí có con cái cũng sẽ ảnh hưởng đến hành vi mua hàng  
# của người dùng.  
data = data.withColumn("Is_Parent", when(data["Children"] > 0, 1).otherwise(0))  
data.show()
```



B5. Kiểm tra sự nhất quán của dữ liệu (int)

Còn 2 biến là: Education và Living_with đang vẫn còn là String

```
[ ] column_types = data.dtypes  
  
categorical_cols = [col_name for col_name, data_type in column_types if data_type == 'string']  
  
print("Categorical variables in the dataset:", categorical_cols)  
  
⇒ Categorical variables in the dataset: ['Education', 'Living_With']
```

-> Thực hiện mã hóa dữ liệu bằng phương pháp **StringIndexer**



B6. Mã hóa dữ liệu: StringIndexer

1.1 StringIndexer

- StringIndexer là một lớp quan trọng trong thư viện `pyspark.ml.feature` của PySpark.
- Được sử dụng để **biến đổi dữ liệu danh mục (categorical data) thành dữ liệu số (numerical data)**.
- Giúp các thuật toán học máy và mô hình thống kê có thể xử lý và phân tích dữ liệu hiệu quả hơn.
- StringIndexer hoạt động bằng cách **gán một chỉ số số duy nhất cho mỗi giá trị duy nhất trong biến dữ liệu danh mục**.



B6. Mã hóa dữ liệu: StringIndexer

1.1

StringIndexer

```
[ ] from pyspark.ml.feature import StringIndexer  
  
# Initialize a StringIndexer for each categorical column  
indexers = [StringIndexer(inputCol=col, outputCol=col+"_index").fit(data) for col in categorical_cols]  
  
# Apply the StringIndexers to the DataFrame  
for indexer in indexers:  
    data = indexer.transform(data)  
  
# Drop the original categorical columns  
data = data.drop(*categorical_cols)  
  
print("All features are now numerical")
```

→ All features are now numerical

```
[ ] # Đổi tên 2 cột phân loại vừa đổi từ chữ sang số  
  
data = data.withColumnRenamed("Education_index", "Education")  
data = data.withColumnRenamed("Living_With_index", "Living_With")
```



B7. Chuẩn hóa dữ liệu



1.2 Chuẩn hóa bằng z-score

Phương pháp này đưa các giá trị dữ liệu về một phân phối chuẩn có **trung bình bằng 0 và độ lệch chuẩn bằng 1** bằng cách trừ đi giá trị trung bình của dữ liệu và chia cho độ lệch chuẩn

$$z = \frac{x - \mu}{\sigma}$$

- x là giá trị dữ liệu cần chuẩn hóa
- μ là giá trị trung bình của tập dữ liệu
- σ là độ lệch chuẩn của tập dữ liệu

Ý nghĩa:

- Z-score > 0 : giá trị lớn hơn giá trị trung bình.
- Z-score < 0 : giá trị nhỏ hơn giá trị trung bình.
- Z-score gần 0: giá trị gần với giá trị trung bình.

Loại bỏ ảnh hưởng của tỷ lệ, biến đổi các giá trị dữ liệu về dạng thống nhất.



1.2 Chuẩn hóa bằng z-score

Xem tên các cột cần chuyển hóa



```
[ ] # Xem tên các cột cần chuyển hóa  
input_features = ds.columns  
print(input_features)  
['Income', 'Kidhome', 'Teenhome', 'Recency', 'Wines', 'Fruits', 'Meat', 'Fish', 'Sweets', 'Gold', 'NumDealsPurchases', 'NumWebPurchases', 'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth', 'CustomerFor', 'Age', 'Total_Spent', 'Children', 'Is_Parent', 'Education', 'Living_With']
```

Tính toán giá trị trung bình và độ lệch chuẩn cho mỗi đặc trưng
Tạo một mảng chứa giá trị trung bình và độ lệch chuẩn của mỗi
đặc trưng

Tính toán giá trị trung bình và độ lệch chuẩn cho mỗi đặc trưng

```
summary = ds.select(  
    [mean(c).alias(c + '_mean') for c in ds.columns] +  
    [stddev(c).alias(c + '_stddev') for c in ds.columns]  
).collect()[0]  
  
# Tạo dictionary để lưu trữ giá trị trung bình và độ lệch chuẩn  
means = {col_name: summary[col_name + '_mean'] for col_name in ds.columns}  
stddevs = {col_name: summary[col_name + '_stddev'] for col_name in ds.columns}  
print(means)  
print(stddevs)
```

B7. Chuẩn hóa dữ liệu



1.2 Chuẩn hóa bằng z-score



B7. Chuẩn hóa dữ liệu

```
[ ] # Kết quả như ý muốn nên chạy cho toàn bộ dataset
for col_name in ds.columns:
    ds = ds.withColumn(col_name, (col(col_name) - means[col_name]) / std devs[col_name])
```

```
# Hiển thị kết quả
ds.show()
```

Income	Kidhome	Teenhome	Recency	Wines	Fruits	Meat	Fish	Sweets	Gold	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases
0.2870399675732395	-0.8225675474319158	-0.9294884884165328	0.31028306977101366	0.9774386452945779	1.5516896492522625	1.6899111229188963	2.452917269820714	1.4833771084606995	0.8523827987390509	0.35095056426763716	1.4265419588890713	2.5030413142030516	-0.5556886393890942
-0.26082305243208287	1.039785990567206	0.9078917943157019	-0.3807274012418051	-0.8724207215113441	-0.6373171524975626	-0.7180673832450953	-0.6508565079385313	-0.6338752946618547	-0.1471506042888537	-0.03724562756562...	-0.68827654170293	1.4265419588890713	-0.229626980874772898
0.912989958634275	-0.8225675474319158	-0.9294884884165328	-0.7953336838494963	0.3578543381432354	0.5784017381238394	-0.17850182908612686	1.3392102083894557	-0.1471506042888537	-0.03724562756562...	-0.68827654170293	1.4265419588890713	-0.5712104276165766	-1.1708954513954386
-1.1707688875458241	0.8225675474319158	0.9078917943157019	-0.5189294954463687	-0.208368641119474	0.9729867016632282	-0.01351070921933...	0.22550314659819687	0.5342639622333476	-0.1686629887176464	-0.7614928126786805	-0.9127939544854241	-0.5556886393890942	-0.1686629887176464
0.29424013444638125	1.039785990567206	-0.9294884884165328	1.5541019175940873	-0.39216877051365284	0.4194447517954563	-0.21863480418886005	0.15247317571680286	-0.00113319717695...	-0.5594188324096811	1.3901776702382043	0.33252457318519545	0.11195662612111859	0.05951820261725025
0.4902708415155483	-0.8225675474319158	0.9078917943157019	-1.1408389193559056	0.6365190504505378	0.3942837540741939	-0.3078191933060449	-0.6873714935592283	0.36391032060279727	-0.5787585827076089	-0.1686629887176464	0.6971970350331541	0.45354015289896614	1.28993188662299392
0.1707688875458241	-0.8225675474319158	0.9078917943157019	-0.5189294954463687	-0.208368641119474	0.9729867016632282	-0.01351070921933...	0.22550314659819687	0.5342639622333476	-0.1686629887176464	-0.7614928126786805	-0.9127939544854241	-0.5556886393890942	-0.1686629887176464
-0.8595982201334206	1.039785990567206	-0.9294884884165328	-0.5880305425456507	-0.6797270374698066	-0.410866817300620143	-0.49510641045213316	-0.6325990151281828	-0.6338752946618547	-0.404700847270328585	-0.1686629887176464	-0.8321478882276323	-0.9127939544854241	-0.5556886393890942
-1.0037409156630877	1.039785990567206	-0.9294884884165328	-1.0371873487039829	-0.8635271668632387	-0.6624781502188251	-0.63780143039629	-0.6325990151281828	-0.5852028256245545	-0.810835563482743	-0.68827654170293	-0.3968203507507219	-0.9127939544854241	-1.1708954813954386
-2.1512617044708418	1.039785990567206	0.9078917943157019	0.65578830277469	-0.8220239118387469	-0.6624781502188251	-0.7180673832450953	-0.6338752946618547	-0.5908098331053681	-0.68827654170293	-1.121652746063393	-0.9127939544854241	-0.8321478882276323	-0.9127939544854241
-2.0652313235097512	-0.8225675474319158	-0.9294884884165328	0.344832359332126546	-0.8872433125915198	-0.2599861866786273	-0.6957712859657991	-0.486539072464339475	-0.5400790855117534	-0.68827654170293	-0.7614928126786805	-0.9127939544854241	-0.8632920863922665	-0.9127939544854241
0.5144258826934246	-0.8225675474319158	-0.9294884884165328	1.1394956349863963	-0.32991388797691507	0.8723427107781788	1.395602638832186	3.420564388769185	0.20674467369083005	-0.2693226083407636	-0.68827654170293	-0.45354015289896614	0.6747250446235947	-0.34354015289896614
0.3435264380969621	1.039785990567206	0.9078917943157019	0.13753045261780897	-0.2142976709277545494	-0.6121561547763003	-0.5084840668197109	-0.6325990151281828	-0.526530314659819687	-0.5787585827076089	-0.6917970350331541	-0.512104276165766	-0.248805213838921999	-0.512104276165766
-1.608926555159317	-0.8225675474319158	-0.9294884884165328	-0.3807274012418051	-0.8961268672396252	-0.31022418212115205	-0.6690156962306437	-0.5778265366971373	-0.6338752946618547	-0.7528163128895956	-0.68827654170293	-0.8321478882276323	-0.9127939544854241	-0.8632920863922665
1.4326561961108994	-0.8225675474319158	-0.9294884884165328	-0.8989825458014192	0.20772490344361151	-0.10893620035105318	-0.2320124265564378	0.3898205822513334	0.9966524108876984	0.020773617268153954	-0.68827654170293	1.0618694969611127	1.1367072067276613	1.905138728632838
-0.4695814390612461	1.039785990567206	0.9078917943157019	0.068424094940961527	-0.747791095642678686	-0.536673161215213	-0.6600975303189252	-0.6508565079385103	-0.3418404804380541	-0.7721560666887303	-0.396827654170293	-0.9127939544854241	-0.8632920863922665	-0.9127939544854241
-0.6595729391309477	-0.8225675474319158	-0.9294884884165328	-0.262363825153342	-0.65608108917407795	-0.536673161215213	-0.5735272427360575996	0.2051252427360575996	-0.3661767169767041	-0.308802105136161924	-0.186866298872276323	-0.5712104276165766	-0.05951820261725025	-0.9127939544854241
1.1629983335426266	-0.8225675474319158	0.9078917943157019	1.45040503469421646	2.0950620127398225	1.3504016674821635	1.4758685890376526	-0.687371493592283	-0.268821776821099	2.5542806577567005	-0.1686629887176464	2.520553446729475	0.45354015289896614	0.98232846526767
-0.8429681572911433	1.039785990567206	-0.9294884884165328	-0.2776972918896	-0.89317234902359	-0.23474118896736495	-0.660097530318925	-0.1396467092487732	-0.07414190073290357	-0.09526487275941306	-0.1686629887176464	-0.7614928126786805	-0.8321478882276323	-0.22962690074772898
-0.693018875535366	-0.8225675474319158	-0.9294884884165328	-0.2770753805898823	-0.6508018553087092	-0.6121561547760256	-0.5723955183548185	-0.2656538252819706	-0.419299679702526	-0.5723955183548185	-0.8787928622193747	-0.68827654170293	-0.248805213838921999	-0.248805213838921999

Fish	Sweets	Gold	NumDealsPurchases	NumWebPurchases	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	CustomerFor	Age	Total_Spent
------	--------	------	-------------------	-----------------	---------------------	-------------------	-------------------	-------------	-----	-------------



B8. Giảm chiều dữ liệu

1.3 PCA

- Principal Component Analysis (PCA) là một kỹ thuật toán học **biến đổi** một tập hợp các điểm dữ liệu có **nhiều chiều** thành **tập hợp các điểm có chiều thấp hơn**, trong khi vẫn giữ lại hầu hết các thông tin quan trọng trong dữ liệu gốc.
- Mục tiêu của PCA là tìm ra các **trục chính** (principal components) sao cho khi chiếu dữ liệu lên các trục này, dữ liệu được trải đều nhất, tức là các **trục này phải chứa nhiều thông tin nhất có thể**.



1.3

PCA



B8. Giảm chiều dữ liệu

PCA procedure

1. Find mean vector 	2. Subtract mean 	3. Compute covariance matrix: $S = \frac{1}{N} \hat{X} \hat{X}^T$
4. Computer eigenvalues and eigenvectors of S: $(\lambda_1, \mathbf{u}_1), \dots, (\lambda_D, \mathbf{u}_D)$ Remember the orthonormality of \mathbf{u}_i .	5. Pick K eigenvectors w. highest eigenvalues 	6. Project data to selected eigenvectors.
7. Obtain projected points in low dimension. 		



1.3

PCA

Dữ liệu trước được sử dụng phải chuyển về dạng vector.

```
[ ] # Lấy tên của các features  
feature_columns = data.columns  
# Lấy giá trị của từng dòng sau đó kết hợp thành 1 vector  
assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")  
assembled_df = assembler.transform(data)
```

```
[ ] assembled_df.show()
```

```
[ ] # Sử dụng PCA trên cột features, kết quả trả về là một vector đã được giảm chiều  
# Xác định số chiều được giảm về là 3  
pca = PCA(k=3, inputCol="features", outputCol="pca_features")  
# fit model PCA với dữ liệu đã được tổng hợp từ trước  
pca_model = pca.fit(assembled_df)  
# Transform dataframe trước đó sử dụng pca model, kết quả trả về cột thuộc tính "pca_feature"  
pca_result = pca_model.transform(assembled_df)
```

- Lấy giá trị của từng dòng sau đó kết hợp nó thành vector “features”
- Sử dụng PCA trên cột features, kết quả trả về là **một vector đã được giảm chiều**
- Xác định số chiều được giảm về là **3**



1.3

PCA

Tạo hàm triết xuất user-defined function để lọc ra các PCs

```
[ ] # Tạo hàm triết xuất user-defined function để lọc ra các PCs
def locPcs(index):
    # Chỗ này lấy giá trị thứ i của vector rồi trả về kiểu float
    def locIndex(vector):
        return float(vector[index])
        # Chỗ này trả về một udf
    return udf(locIndex, DoubleType())
```

```
[ ] # Sử dụng hàm locPcs để add vào dữ liệu từ trước 3 cột (tương ứng với 3 PCs)
pca_df = pca_result.withColumn("pc1", locPcs(0)(col("pca_features")))
    .withColumn("pc2", locPcs(1)(col("pca_features")))
    .withColumn("pc3", locPcs(2)(col("pca_features")))

[ ] pca_df.show()
```

- Lấy giá trị thứ i của vector rồi trả về kiểu float
- Trả về một hàm do người dùng xác định
- Sử dụng hàm locPcs để add vào dữ liệu từ trước 3 cột (tương ứng với 3 PCs)



1.3

PCA

Trực quan dữ liệu sử dụng thư viện Matplotlib

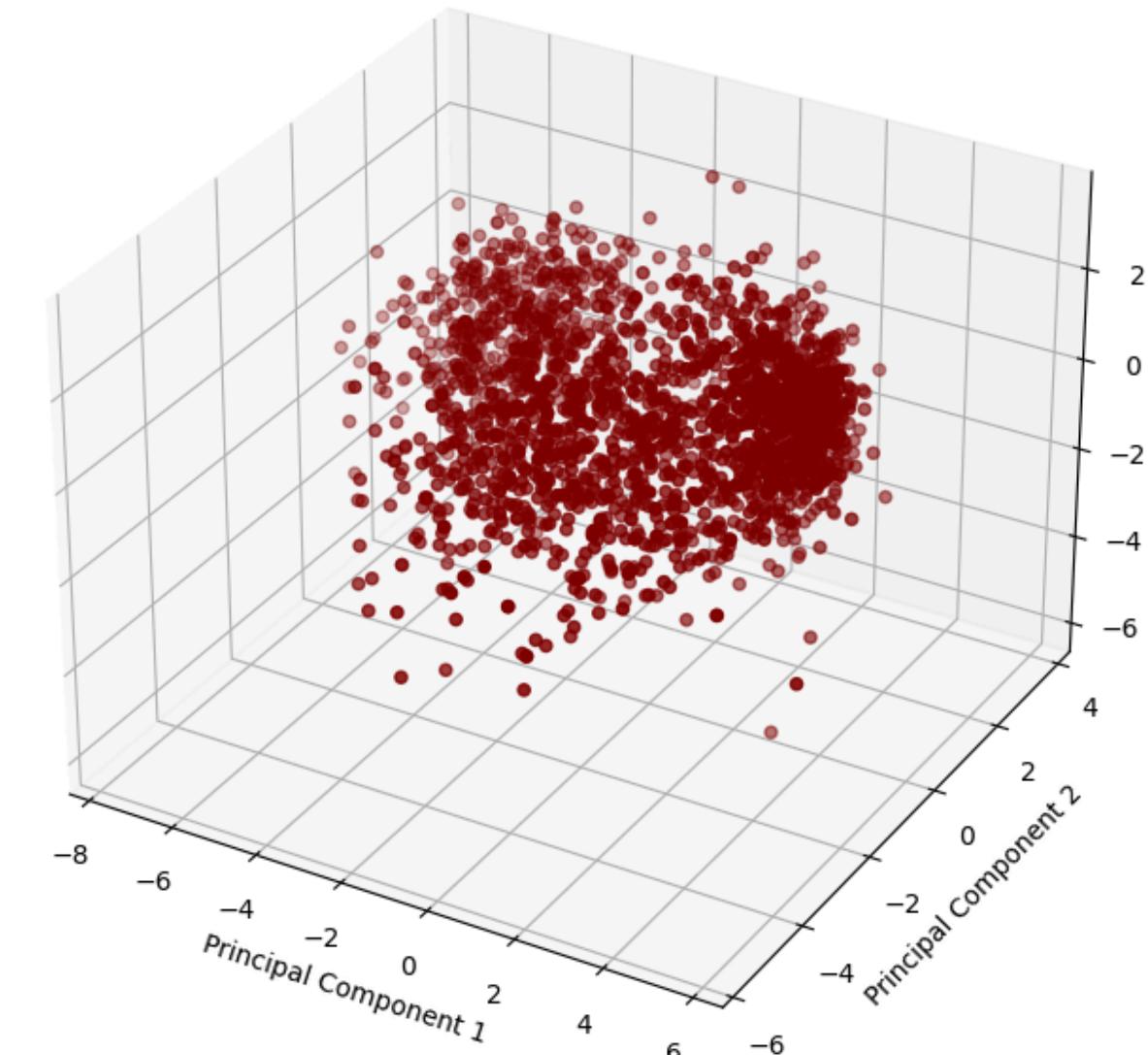
```
[ ] # TẢI VỀ DATASET PCA  
ds_download = pca_df.toPandas()  
# Lưu DataFrame pandas dưới dạng file CSV  
ds_download.to_csv('pca_data.csv', index=False)
```

```
[ ] # Chuyển dữ liệu về dạng pandas dataframe để có thể trực quan hóa  
plot_pca_df = pca_df.toPandas()
```

```
[ ] # Xác định cột x, y, z tương ứng với các PCs  
x = plot_pca_df["pc1"]  
y = plot_pca_df["pc2"]  
z = plot_pca_df["pc3"]
```

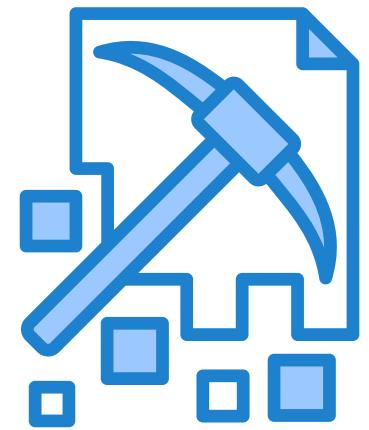
```
▶ # Trực quan dữ liệu sử dụng thư viện Matplotlib  
fig = plt.figure(figsize=(10, 8))  
ax = fig.add_subplot(111, projection="3d")  
ax.scatter(x, y, z, c="maroon", marker="o")  
ax.set_title("A 3D Projection Of Data In The Reduced Dimension")  
ax.set_xlabel('Principal Component 1')  
ax.set_ylabel('Principal Component 2')  
ax.set_zlabel('Principal Component 3')  
plt.show()
```

A 3D Projection Of Data In The Reduced Dimension



- Souce Code:

<https://colab.research.google.com/drive/1JXlY6m9DDooK6Wmn5ttYZQI7FWmw9gG4?usp=sharing>



2. Các thuật toán khai thác dữ liệu

2.1 Khoảng cách Euclidean

- Được đặt tên theo nhà toán học Hy Lạp Euclid.
- Được sử dụng rộng rãi trong nhiều lĩnh vực như xử lý ảnh, nhận dạng mẫu, và **phân tích dữ liệu**.
- Khoảng cách này được sử dụng để **đo lường khoảng cách giữa các điểm trong không gian**.

$$d(P, Q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

- Trong đó: q_i và p_i là các thành phần của điểm P và Q tương ứng trong không gian n chiều



2. Các thuật toán khai thác dữ liệu

2.2 Phương Sai

Đo lường **mức độ phân tán** của các giá trị trong một tập dữ liệu. (Đo độ lệch của mỗi điểm dữ liệu so với giá trị trung bình của toàn bộ tập dữ liệu)

- Đánh giá mức độ phân tán của dữ liệu
- Ước lượng độ chính xác của mô hình
- Ước lượng sự khác biệt giữa các mẫu
- Xác định độ tin cậy của dữ liệu

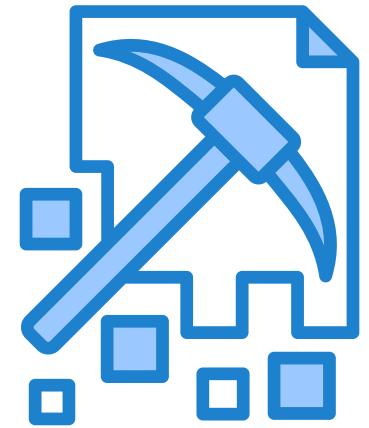
Trong đó:

- $$\sigma^2 = \frac{\sum (x - \mu)^2}{N}$$
- x là giá trị trong tập dữ liệu.
 - μ là giá trị trung bình của tập dữ liệu
 - N là số lượng giá trị trong tập dữ liệu



III. Triển khai thuật toán



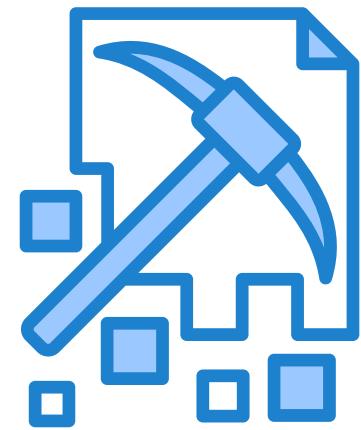


2. Các thuật toán khai thác dữ liệu

2.3 Thuật toán K-Mean

2.2.1. Giới thiệu

- **K-Means Clustering** là thuật toán máy học không có giám sát (unsupervised learning) được sử dụng để **phân chia một tập dữ liệu thành các nhóm** (clusters) sao cho các điểm dữ liệu trong cùng một nhóm có đặc điểm giống nhau và các nhóm khác nhau có đặc điểm khác nhau.
- **Unsupervised machine learning:** là một quá trình dạy máy học các dữ liệu không dán nhãn, chưa được phân loại và khởi tạo trên dữ liệu đó mà không cần có sự giám sát. Không cần dữ liệu đã được đào tạo trước đó, máy học trong trường hợp này chỉ cần sắp xếp dữ liệu thành các cụm dựa theo các điểm tương đồng, mẫu, biến thể.

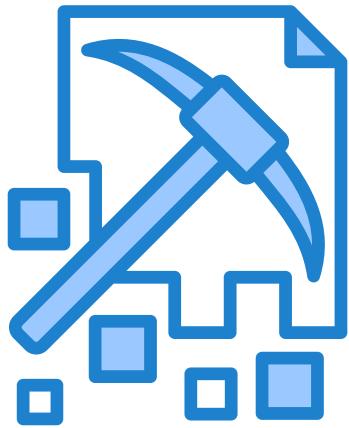


2. Các thuật toán khai thác dữ liệu

2.3 Thuật toán K-Mean

2.2.1. Giới thiệu





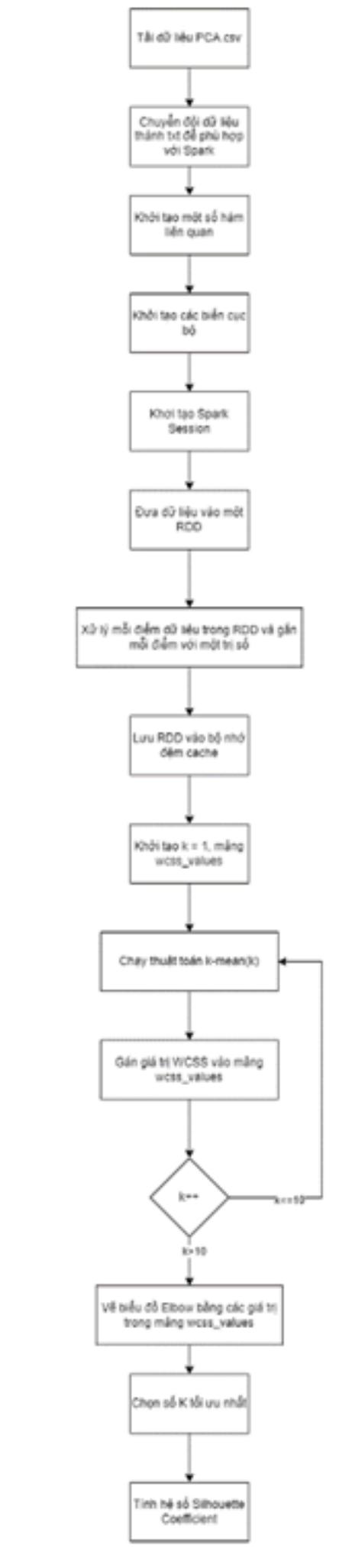
2. Các thuật toán khai thác dữ liệu

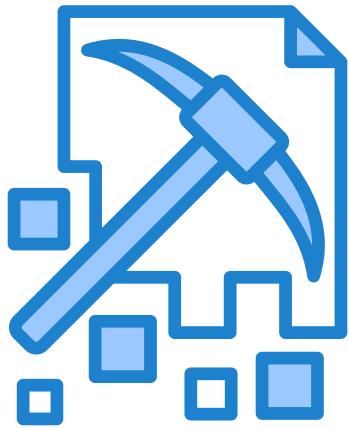
2.2.2. Mục đích

- Phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong **cùng một cụm có tính chất giống nhau**.
- Bằng cách từ dữ liệu đầu vào và số lượng nhóm muốn tìm, chỉ ra **center** của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng.

2.3

Thuật toán K-Mean





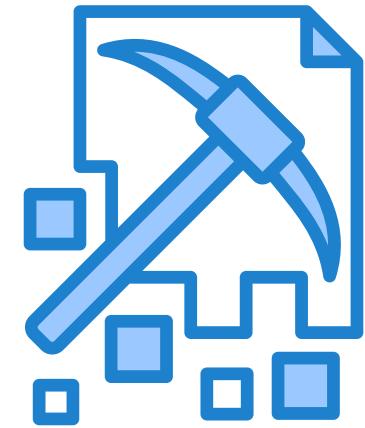
2. Các thuật toán khai thác dữ liệu

2.3 Thuật toán K-Mean

2.2.3 K-means clustering và cách hoạt động

Các bước của thuật toán k-Means:

B1: Khởi tạo ngẫu nhiên k tâm cụm : $\mu_1, \mu_2, \dots, \mu_k, \mu_1, \mu_2, \dots, \mu_k$.



2. Các thuật toán khai thác dữ liệu

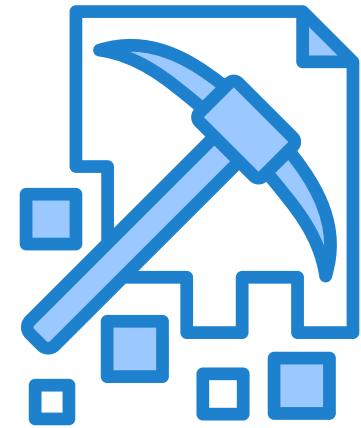
2.3 Thuật toán K-Mean

2.2.3 K-means clustering và cách hoạt động

Các bước của thuật toán k-Means:

B1: Khởi tạo ngẫu nhiên k tâm cụm : $\mu_1, \mu_2, \dots, \mu_k$.

B2: Lặp lại quá trình cập nhật tâm cụm cho tới khi dừng:



2. Các thuật toán khai thác dữ liệu

2.3 Thuật toán K-Mean

2.2.3 K-means clustering và cách hoạt động

Các bước của thuật toán k-Means:

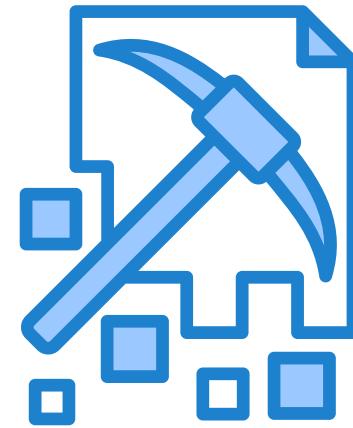
B1: Khởi tạo ngẫu nhiên k tâm cụm : $\mu_1, \mu_2, \dots, \mu_k, \mu_1, \mu_2, \dots, \mu_k$.

B2: Lặp lại quá trình cập nhật tâm cụm cho tới khi dừng:

a) Xác định nhãn cho từng điểm dữ liệu ci dựa vào khoảng cách tới từng tâm cụm

$$c_i = \arg \min_j \|x_i - \mu_j\|_2^2$$

- x_i : điểm dữ liệu thứ i
- $\|x_i - \mu_j\|_2^2$: bình phương của khoảng cách Euclid giữa điểm dữ liệu x_i và tâm cụm μ_j
- c_i : nhãn của điểm dữ liệu x_i , là chỉ số của cụm mà x_i thuộc về, được xác định bằng cách tìm tâm cụm gần nhất.



2. Các thuật toán khai thác dữ liệu

2.3 Thuật toán K-Mean

2.2.3 K-means clustering và cách hoạt động

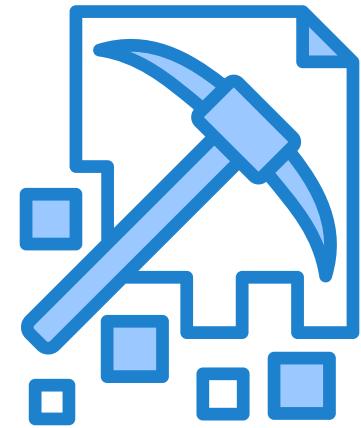
Các bước của thuật toán k-Means:

B1: Khởi tạo ngẫu nhiên k tâm cụm : $\mu_1, \mu_2, \dots, \mu_k, \mu_1, \mu_2, \dots, \mu_k$.

B2: Lặp lại quá trình cập nhật tâm cụm cho tới khi dừng:

b) Tính toán lại tâm cho từng cụm theo trung bình của toàn bộ các điểm dữ liệu trong một cụm:

- μ_j : tâm cụm thứ j
 - $1(c_i=j)$: hàm chỉ thị, trả về giá trị 1 nếu điểm dữ liệu x_i thuộc về cụm j , ngược lại trả về 0.
 - x_i : điểm dữ liệu thứ i
 - Biểu thức tính toán trung bình có nghĩa là lấy tổng các điểm dữ liệu thuộc về cụm j và chia cho số lượng các điểm dữ liệu đó để cập nhật lại tâm cụm μ_j
- $$\mu_j = \frac{\sum_{i=1}^n 1(c_i = j)x_i}{\sum_{i=1}^n 1(c_i = j)}$$



2. Các thuật toán khai thác dữ liệu

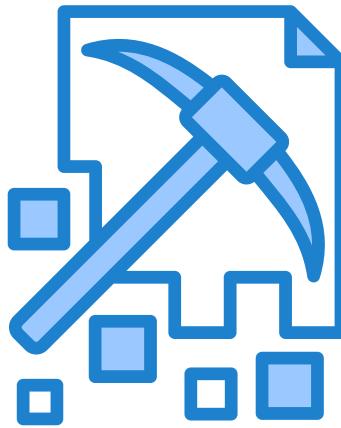
2.3 Thuật toán K-Mean

2.2.3 K-means clustering và cách hoạt động

Tham số mà chúng ta cần chọn chính là **số lượng cụm k** . Thời điểm ban đầu ta sẽ khởi tạo k điểm dữ liệu một cách ngẫu nhiên và sau đó gán các tâm bằng giá trị của k điểm dữ liệu này.

Các bước trong vòng lặp ở bước thứ 2 thực chất là:

- a) **Gán nhãn** cho mỗi điểm dữ liệu bằng với nhãn của tâm cụm gần nhất.
- b) **Dịch chuyển dần dần tâm cụm**. μ_j tới trung bình của những điểm dữ liệu mà được phân về j .

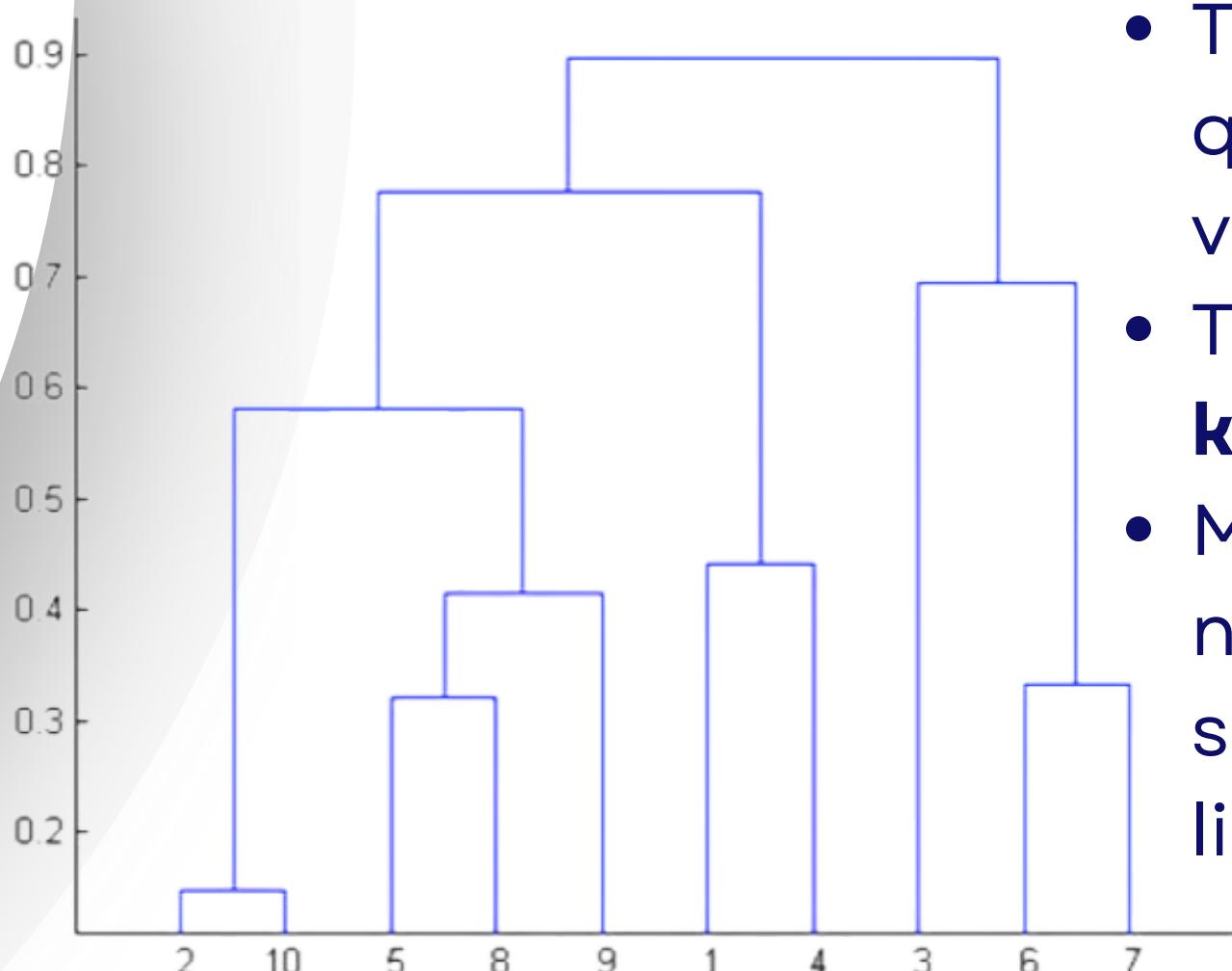


2. Các thuật toán khai thác dữ liệu

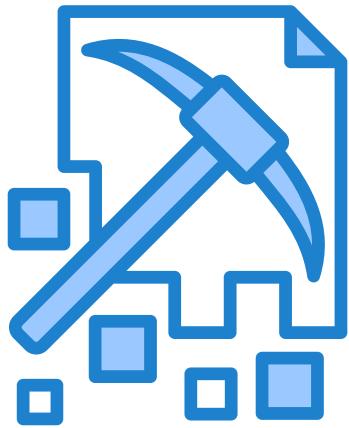
2.4

Thuật toán Hierarchical Clustering

Hierarchical Clustering (Phân cụm phân cấp) là một kỹ thuật phân cụm mà các điểm dữ liệu được sắp xếp thành một **cấu trúc phân cấp giống như cấu trúc cây**, thường được thể hiện bằng một biểu đồ cây gọi là dendrogram.



- Trục x: **chỉ số index** của các quan sát trong nhóm được phân vào một cụm.
- Trục y: là giá trị **thước đo sự khác biệt** giữa các cụm.
- Một cụm được đại diện bởi một node (nút) mà toàn bộ các quan sát khác nếu thuộc cụm thì đều liên kết tới nút đó

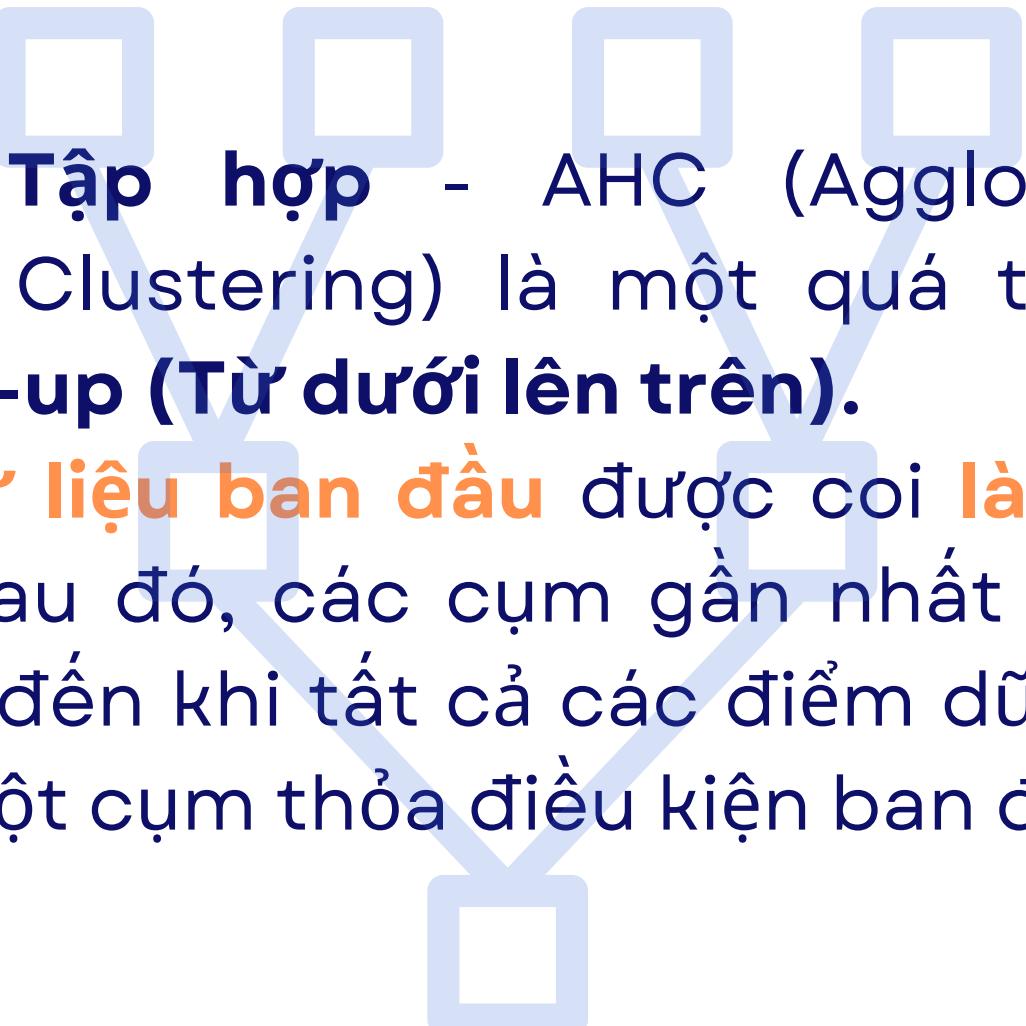


2. Các thuật toán khai thác dữ liệu

2.4

Thuật toán Hierarchical Clustering

2.4.1 Phân cụm Tập hợp



- **Phân cụm Tập hợp** - AHC (Agglomerative Hierarchical Clustering) là một quá trình phân cụm **Bottom-up (Từ dưới lên trên)**.
- Mỗi **điểm dữ liệu ban đầu** được coi **là một cụm** riêng biệt. Sau đó, các cụm gần nhất được **hợp nhất lại** cho đến khi tất cả các điểm dữ liệu được gộp thành một cụm thỏa điều kiện ban đầu đặt ra.



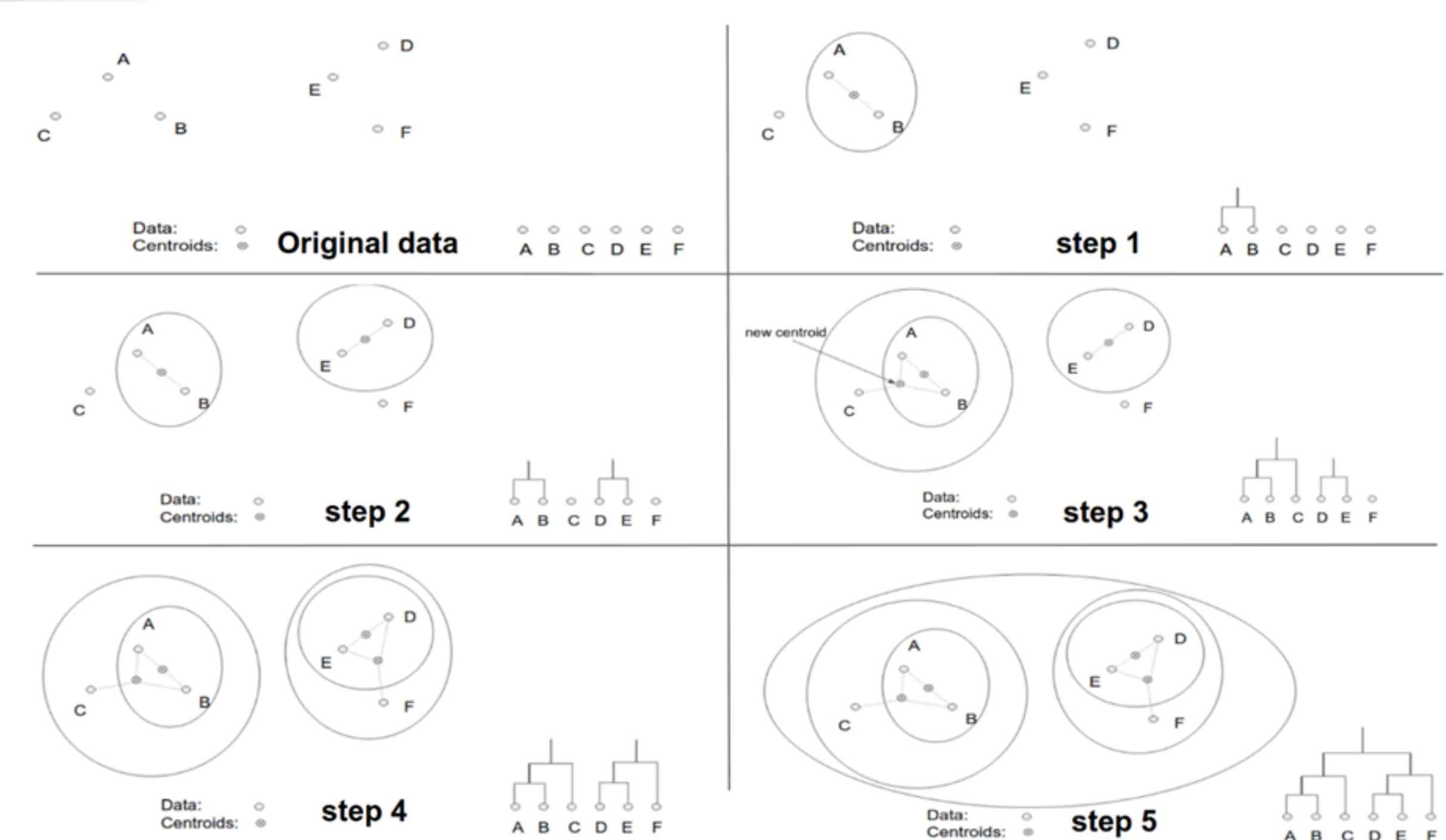
2. Các thuật toán khai thác dữ liệu

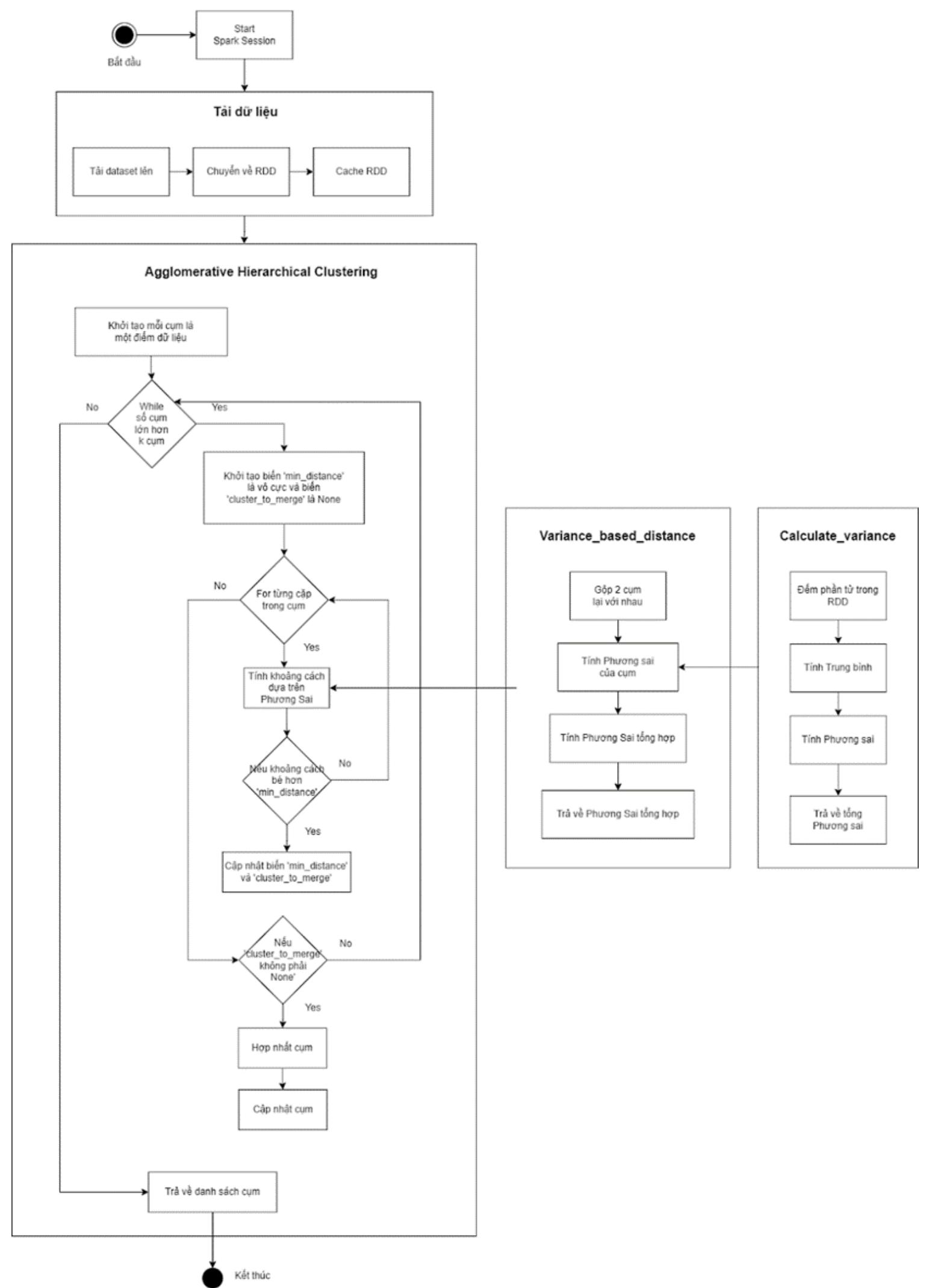


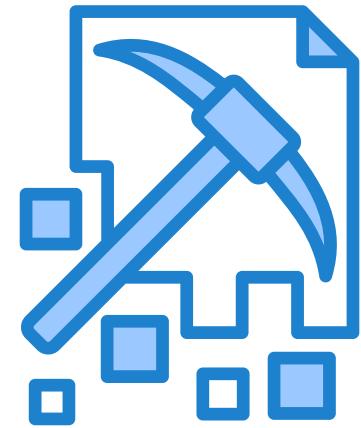
2.4

Thuật toán Hierarchical Clustering

2.4.1 Phân cụm Tập hợp







2. Các thuật toán khai thác dữ liệu

2.4

Thuật toán Hierarchical Clustering

2.4.1 Phân cụm Tập hợp

Các bước cụ thể

B1: Khởi tạo

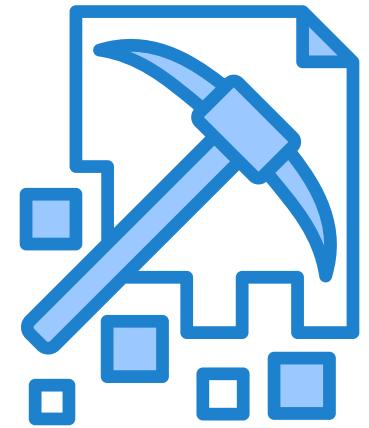
- Mỗi điểm dữ liệu sẽ được xét là một cụm riêng biệt.
- Giả sử ta có N điểm dữ liệu, ban đầu ta sẽ có N cụm.

B2: Tính toán ma trận khoảng cách

- Thực hiện tính toán khoảng cách giữa các cặp cụm dựa trên khoảng cách Euclidean.

B3: Tìm cụm gần nhất

- Dựa trên ma trận khoảng cách từ bước 2, ta thực hiện xét và tìm hai cụm có khoảng cách nhỏ nhất.



2. Các thuật toán khai thác dữ liệu

2.4

Thuật toán Hierarchical Clustering

2.4.1 Phân cụm Tập hợp

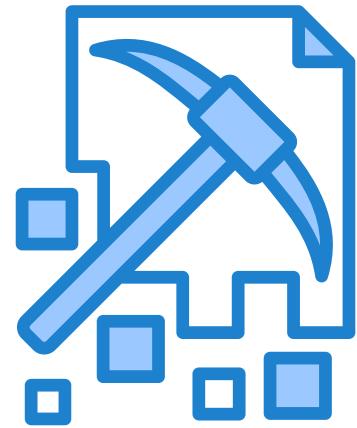
B4: Hợp nhất cụm gần nhất

- Hợp nhất hai cụm gần nhất thành một cụm mới.
- Cập nhật ma trận khoảng cách để phản ánh sự hợp nhất này. Khoảng cách giữa cụm mới và các cụm còn lại được tính toán lại tùy theo phương pháp đo khoảng cách mà ta sử dụng.

B5: Lặp lại các bước 3 và 4 cho đến khi tất cả các điểm dữ liệu được hợp nhất thành số cụm k thỏa điều kiện đặt ra.



2. Các thuật toán khai thác dữ liệu

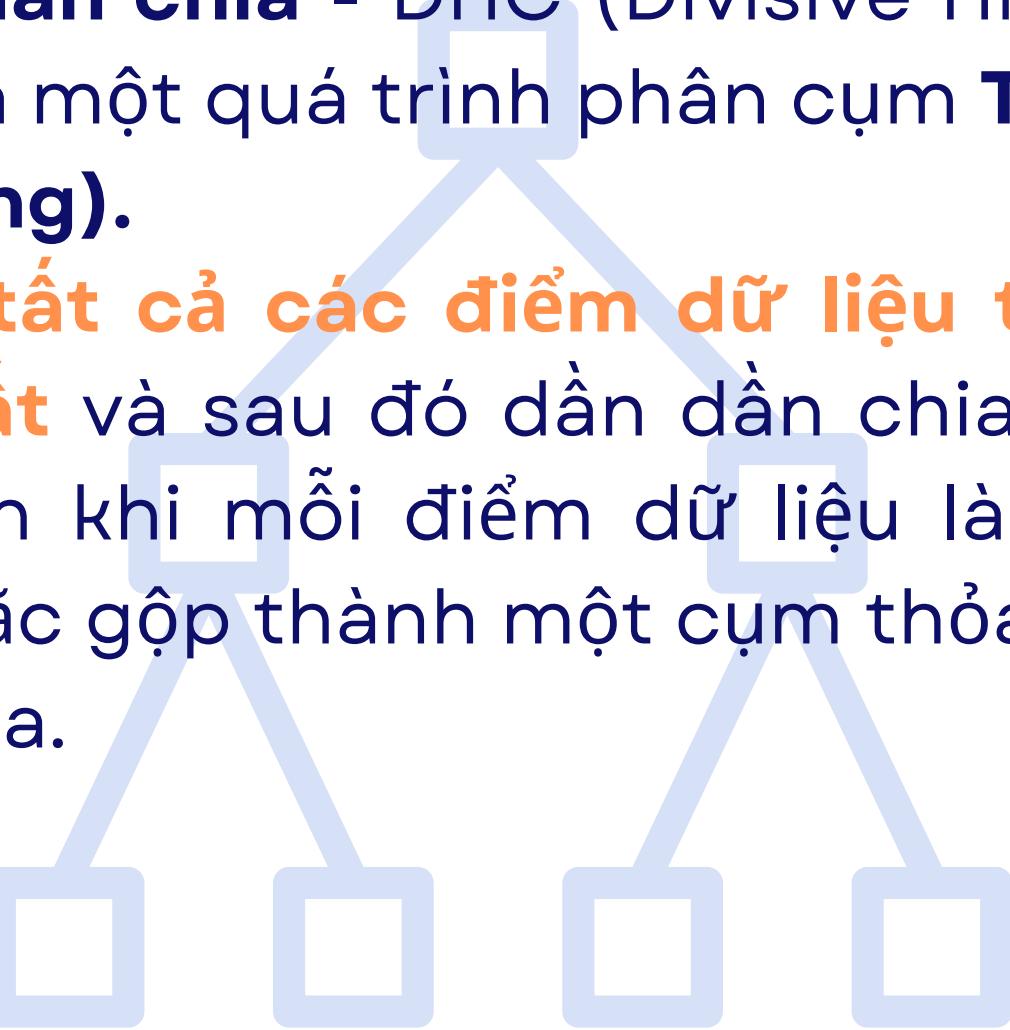


2.4

Thuật toán Hierarchical Clustering

2.4.2 Phân cụm Phân chia

- **Phân cụm Phân chia** - DHC (Divisive Hierarchical Clustering) là một quá trình phân cụm **Top-down (Từ trên xuống)**.
- Bắt đầu với **tất cả các điểm dữ liệu trong một cụm duy nhất** và sau đó dần dần chia **tách các cụm** cho đến khi mỗi điểm dữ liệu là một cụm riêng biệt hoặc gộp thành một cụm thỏa điều kiện ban đầu đặt ra.





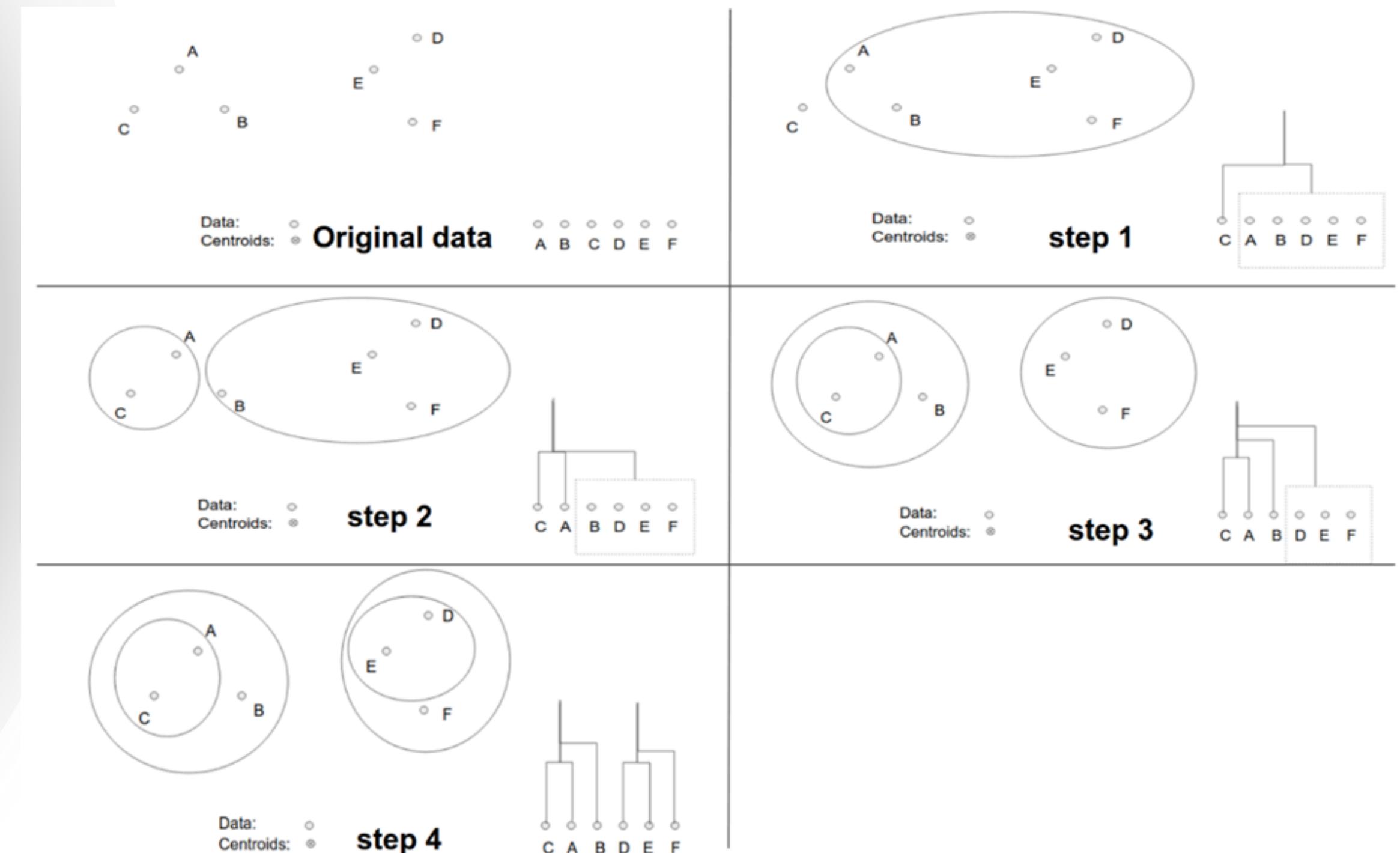
2. Các thuật toán khai thác dữ liệu

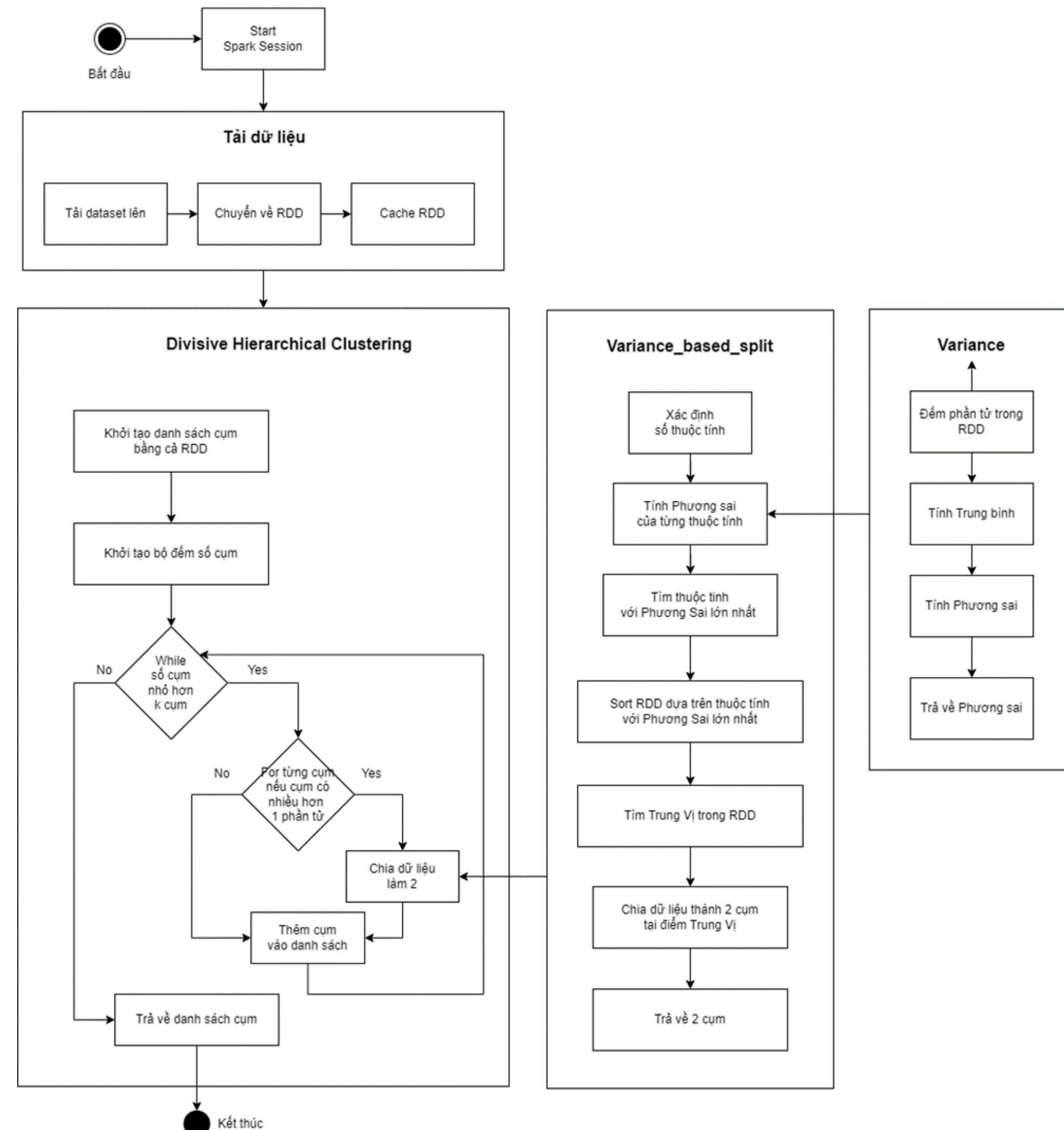


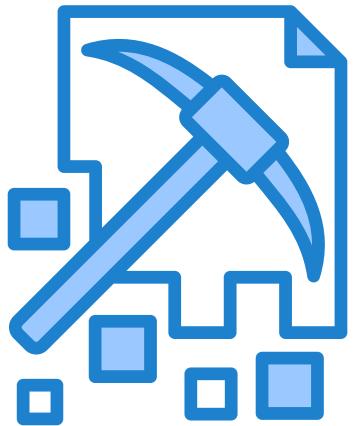
2.4

Thuật toán Hierarchical Clustering

2.4.2 Phân cụm Phân chia







2. Các thuật toán khai thác dữ liệu

2.4

Thuật toán Hierarchical Clustering

2.4.1 Phân cụm Phân chia

Các bước cụ thể

B1: Khởi tạo

- Ta xét tất cả các điểm dữ liệu ban đầu là một cụm duy nhất

B2: Chọn cụm chia tách

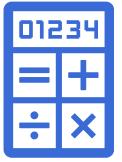
- Chọn cụm lớn nhất hoặc cụm có độ biến thiên lớn nhất để chia tách.

B3: Phân chia cụm

- Sử dụng phương pháp phân chia khoảng cách đầy đủ (**Complete - linkage**) để chia cụm đã chọn thành hai cụm con.

- **Cập nhật ma trận khoảng cách** để phản ánh sự phân chia cụm thường bao gồm tính toán lại khoảng cách giữa các cụm mới và các cụm còn lại.

B4: Lặp lại các bước 2 và 3 cho đến đạt được số cụm k thỏa điều kiện đặt ra



Là số liệu đo lường mức độ phù hợp của mỗi điểm dữ liệu với cụm được chỉ định của nó.

Nó kết hợp thông tin về cả **sự gắn kết** (mức độ gần của một điểm dữ liệu với các điểm khác trong cụm riêng của nó) và **sự phân tách** (khoảng cách của một điểm dữ liệu với các điểm trong các cụm khác) của điểm dữ liệu.



3. Phương pháp đánh giá

Silhouette Coefficient



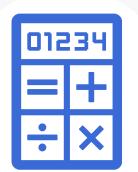
Hệ số nằm trong khoảng từ **-1 đến 1**

Trong đó:

Giá trị gần 1 cho biết điểm dữ liệu được phân cụm tốt

Giá trị gần 0 cho thấy các cụm chồng chéo

Giá trị gần -1 cho biết điểm dữ liệu được phân loại sai



Công thức:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}$$



Trong đó:

$a(i)$: Khoảng cách trung bình của điểm đó với tất cả các điểm khác trong cùng một cụm.

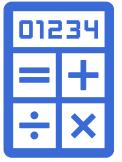
$b(i)$: Khoảng cách trung bình của điểm đó với tất cả các điểm trong cụm gần nhất với cụm của nó.

$s(i)$: Hệ số Silhouette Coefficient điểm thứ i



3. Phương pháp đánh giá

Silhouette Coefficient



Chỉ số Calinski–Harabasz (CHI), còn được gọi là Tiêu chí Tỷ lệ Phương sai (VRC), là một thước đo để đánh giá các thuật toán phân cụm, được Tadeusz Caliński và Jerzy Harabasz giới thiệu vào năm 1974. Đây là một thước đo đánh giá nội bộ, trong đó việc đánh giá chất lượng phân cụm chỉ dựa trên tập dữ liệu và kết quả phân cụm, không dựa trên các nhãn chân lý gốc bên ngoài



3. Phương pháp đánh giá

Calinski-Harabasz Index



Hệ số nằm trong khoảng từ **-1 đến 1**

Trong đó:

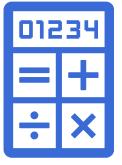
Giá trị gần 1 cho biết điểm dữ liệu được phân cụm tốt

Giá trị gần 0 cho thấy các cụm chồng chéo

Giá trị gần -1 cho biết điểm dữ liệu được phân loại sai



Công thức:



Cho một tập dữ liệu gồm n điểm: $\{x_1, \dots, x_n\}$, và việc phân loại các điểm này vào k cụm: $\{C_1, \dots, C_k\}$, chỉ số Calinski–Harabasz (CH) được định nghĩa là tỷ số giữa sự phân tán giữa các cụm ($BCSS$) và sự phân tán trong cụm ($WCSS$), được chuẩn hóa theo số bậc tự do của chúng

$$CH = \frac{BCSS/(k - 1)}{WCSS/(n - k)}$$

Từ công thức trên, chúng ta có thể kết luận rằng các giá trị lớn của chỉ số Calinski–Harabasz thể hiện sự phân cụm tốt hơn.



3. Phương pháp đánh giá

Calinski–Harabasz Index



BCSS (Between-Cluster Sum of Squares) là tổng trong số của các khoảng cách Euclid bình phương giữa mỗi tâm cụm (trung bình) và tâm dữ liệu tổng thể (trung bình):

$$BCSS = \sum_{i=1}^k n_i \|c_i - c\|^2$$

Trong đó n_i là số điểm trong cụm C_i , c_i là tâm của C_i , và c là tâm tổng thể của dữ liệu. $BCSS$ đo lường mức độ phân tách của các cụm (càng cao càng tốt).



WCSS (Within-Cluster Sum of Squares) là tổng các khoảng cách Euclid bình phương ra các điểm dữ liệu và tâm cụm tương ứng của chúng:

$$WCSS = \sum_{i=1}^k \sum_{x \in C_i} \|x - c_i\|^2$$

$WCSS$ đo lường độ chât chẽ hoặc sự liên kết của các cụm (càng nhỏ càng tốt). Việc tối thiểu hóa $WCSS$ là mục tiêu của các thuật toán phân cụm dựa trên tâm, chẳng hạn như k-means.



IV. Kết quả đạt được và kết luận





IV. Kết quả đạt được và kết luận

Chú ý:

- Bài toán nhóm sử dụng là **bài toán học máy không giám sát**.
- **Không có các nhãn** được gắn thẻ từ trước để đánh giá hoặc cho điểm cho từng mô hình.
- Nên mục đích chính của bài toán nhóm đem đến là để nghiên cứu các mẫu dữ liệu có trong cụm được xác định bởi các thuật toán nhóm chuẩn bị (K-Mean, Divisive Hierarchical Clustering (DHC), Agglomerative Hierarchical Clustering (AHC)).
- Để xác định được sự hình thành và bản chất của dữ liệu tập khách hàng thì nhóm sẽ xét dữ liệu thông qua việc trực quan hóa dữ liệu và đưa ra kết luận



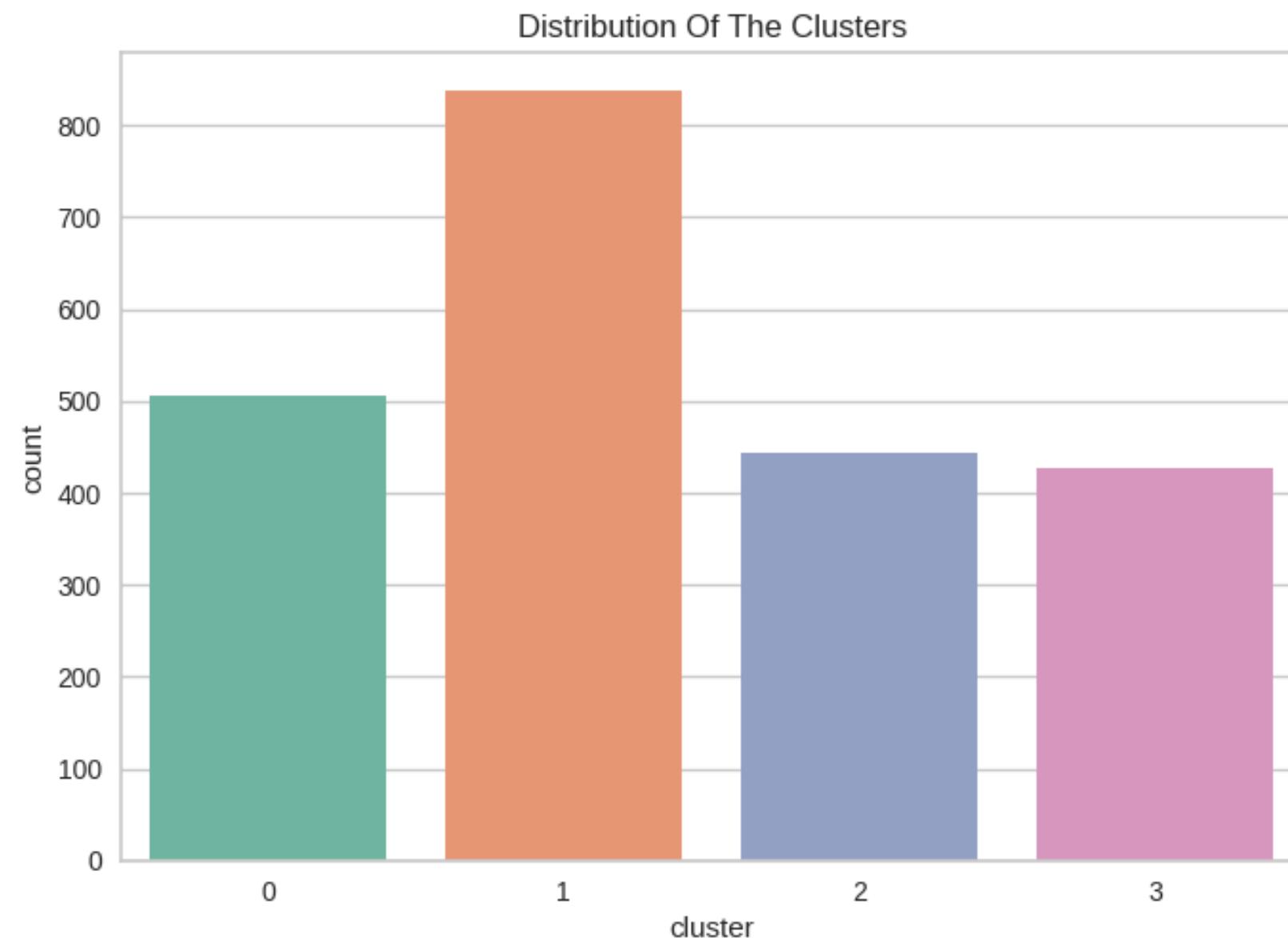
Thuật toán K-means





IV. Kết quả đạt được và kết luận

1. Xem sự phân bố phân tử ở các cụm với nhau



Nhận xét: Có vẻ như Cụm 2 ($k = 1$) chiếm gấp đôi số lượng phân tử, còn 3 cụm còn lại thì có sự phân bố đồng đều với nhau.



IV. Kết quả đạt được và kết luận

2. Xem xét sự phân bố thu nhập và chi tiêu ở các cụm với nhau





IV. Kết quả đạt được và kết luận

2. Xem xét sự phân bố thu nhập và chi tiêu ở các cụm với nhau

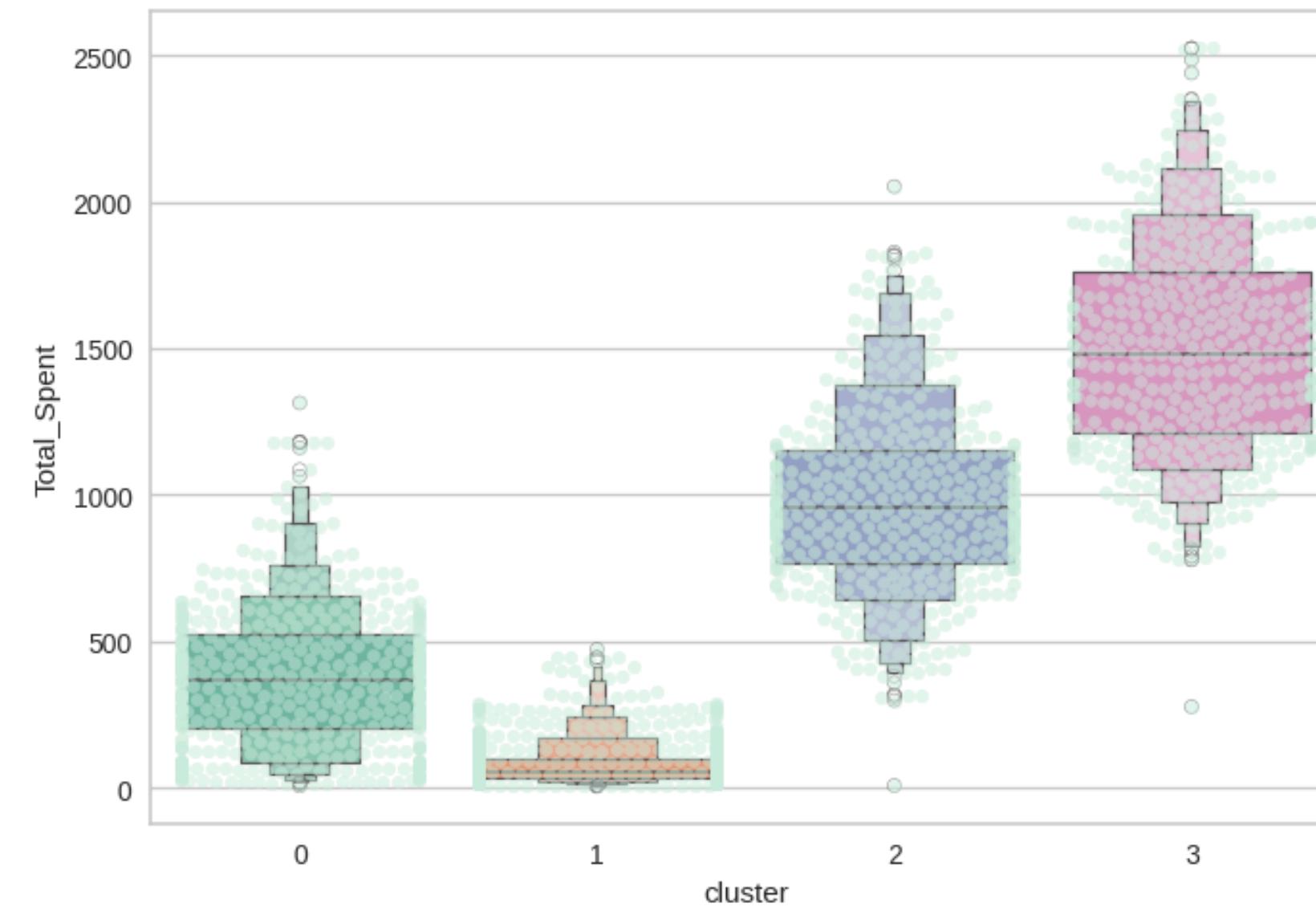
Nhận xét:

- Nhóm 1 ($k = 0$): Là nhóm khách hàng thu nhập trung bình và chi tiêu ít.
- Nhóm 2 ($k = 1$): Là nhóm khách hàng thu thấp và chi tiêu cũng thấp
- Nhóm 3 ($k = 2$): Nhóm khách hàng có thu nhập trung bình và chi tiêu cao.
- Nhóm 4 ($k = 3$): Nhóm khách hàng có thu nhập cao và chi tiêu cao.



IV. Kết quả đạt được và kết luận

3. Xem xét sự phân bố chi tiết các cụm theo các sản phẩm khác nhau trong dữ liệu

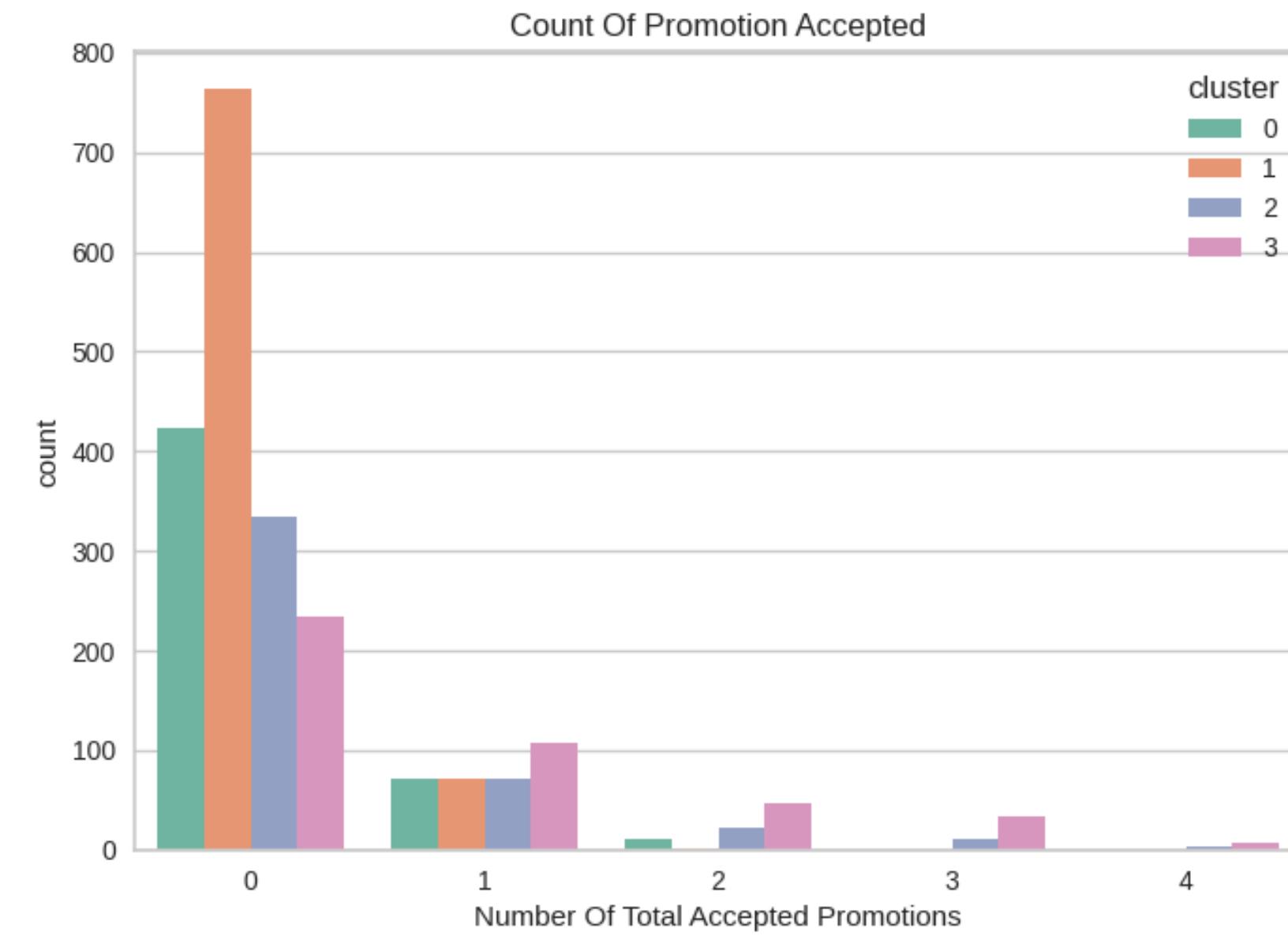


Nhận xét: Nhóm khách hàng Cụm 3 là nhóm khách hàng lớn nhất. Theo sau đó là nhóm khách hàng thuộc Cụm 2.



IV. Kết quả đạt được và kết luận

4. Xem xét sự phân bố số lượng các chiến dịch quảng cáo và số ủng hộ





IV. Kết quả đạt được và kết luận

4. Xem xét sự phân bố số lượng các chiến dịch quảng cáo và số ủng hộ

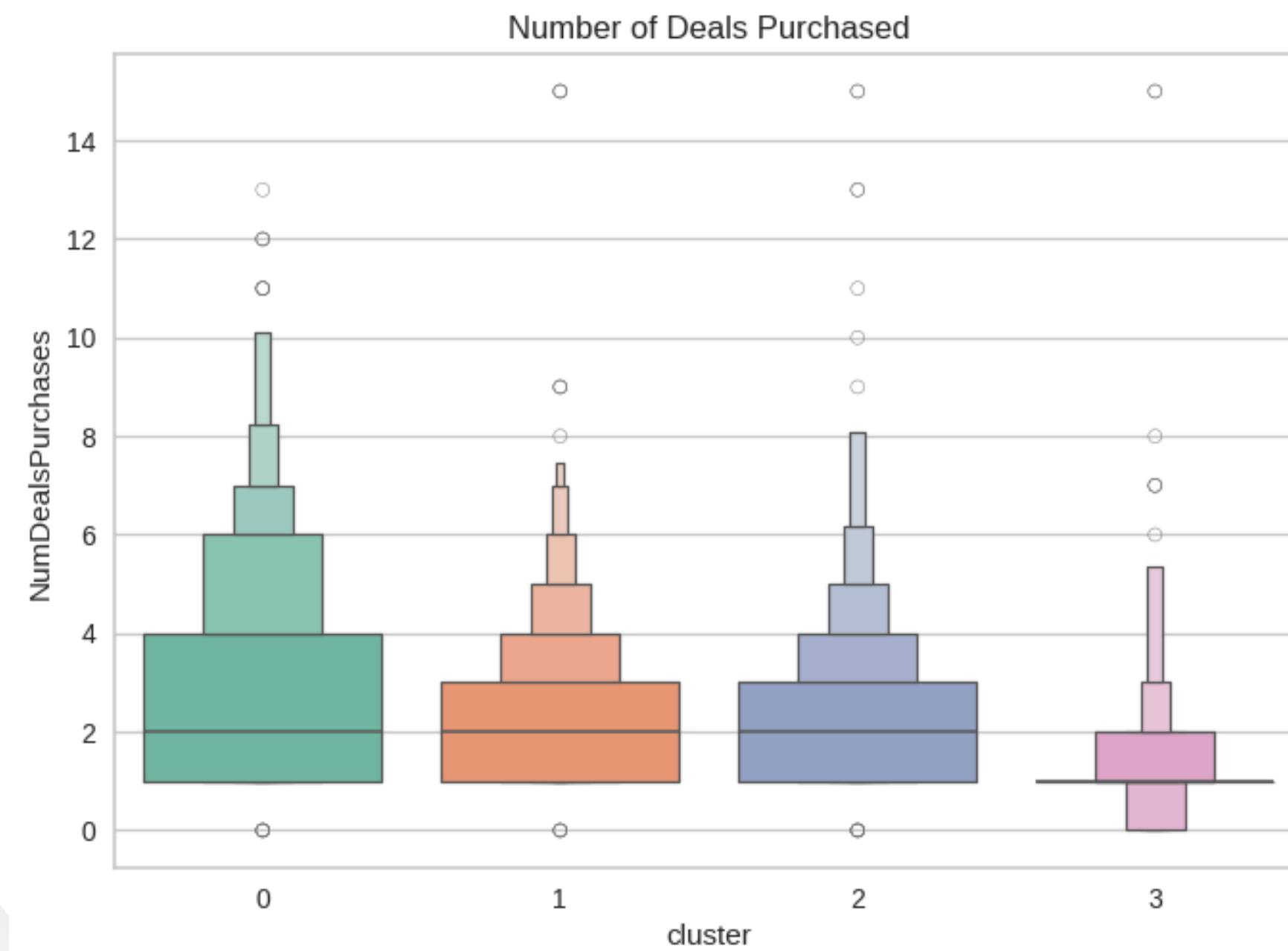
Nhận xét:

- Cho đến nay vẫn chưa có Cụm áp đảo nào dựa trên các chiến dịch.
- Nhìn chung rất ít người tham gia. Hơn nữa, không có Cụm nào rải đều cả 4 chiến dịch..
- Có lẽ cần phải có các chiến dịch được nhắm mục tiêu tốt hơn và được lên kế hoạch tốt hơn để tăng doanh số bán hàng.



IV. Kết quả đạt được và kết luận

5. Xem xét số lượng chốt đơn hàng giảm giá ở các tập khách hàng





IV. Kết quả đạt được và kết luận

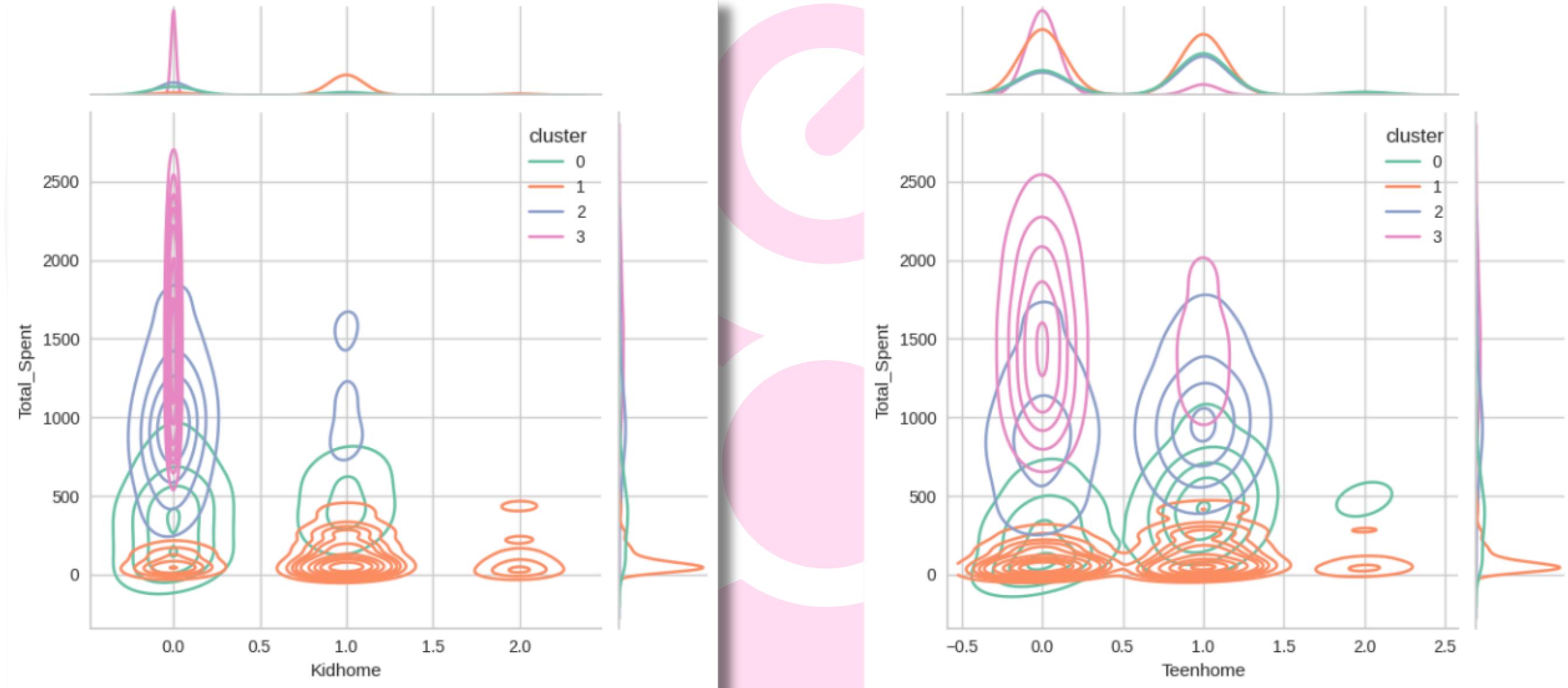
5. Xem xét số lượng chốt đơn hàng giảm giá ở các tập khách hàng

Nhận xét: Không giống như các chiến dịch, các hoạt động mua hàng giảm giá đều hoạt động tốt. Nó mang lại kết quả tốt nhất với cụm 0. Tuy nhiên, các khách hàng cụm 3 không quan tâm nhiều đến các hoạt động này. Nhưng có vẻ các khách hàng ở cụm 2 cũng khá thích giảm giá.



IV. Kết quả đạt được và kết luận

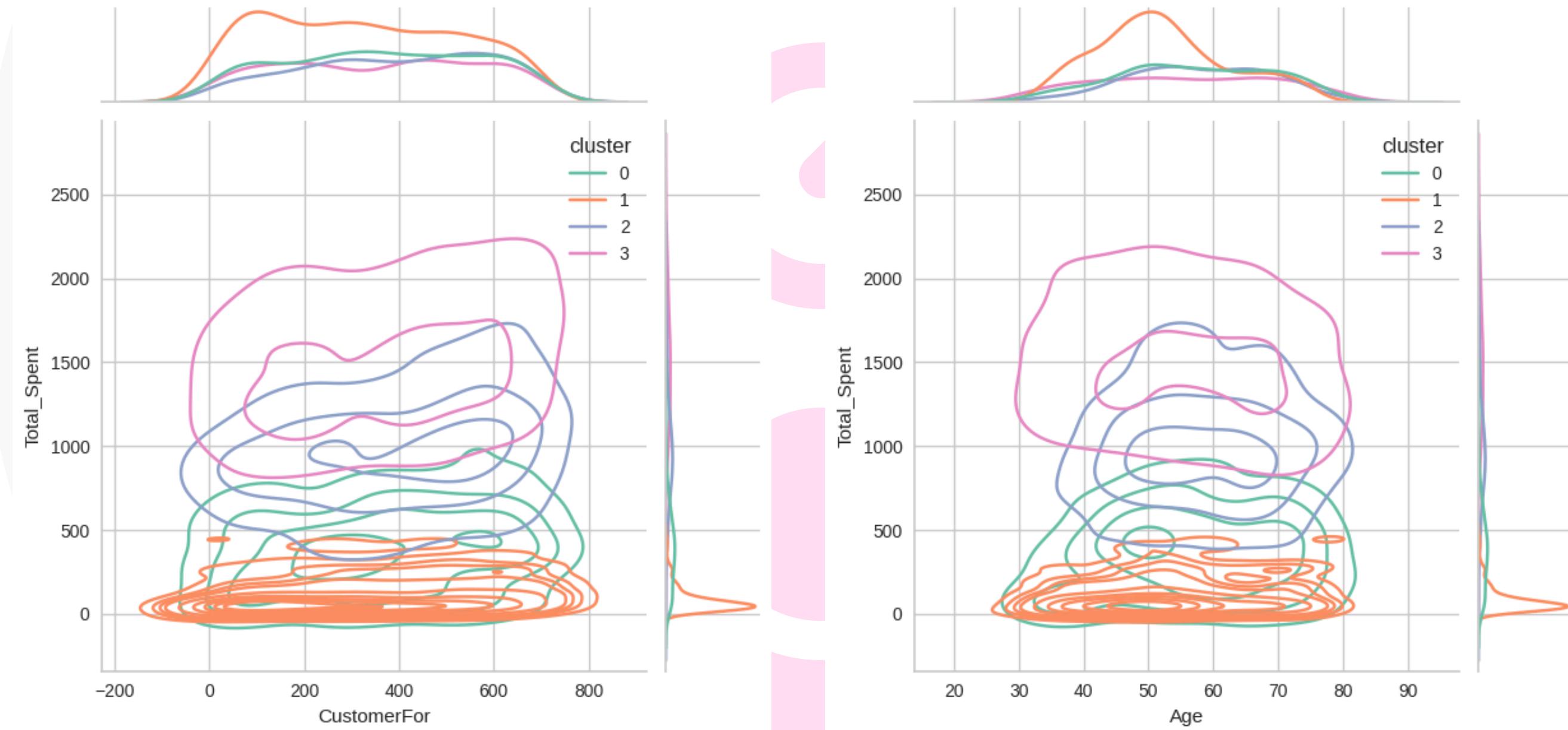
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

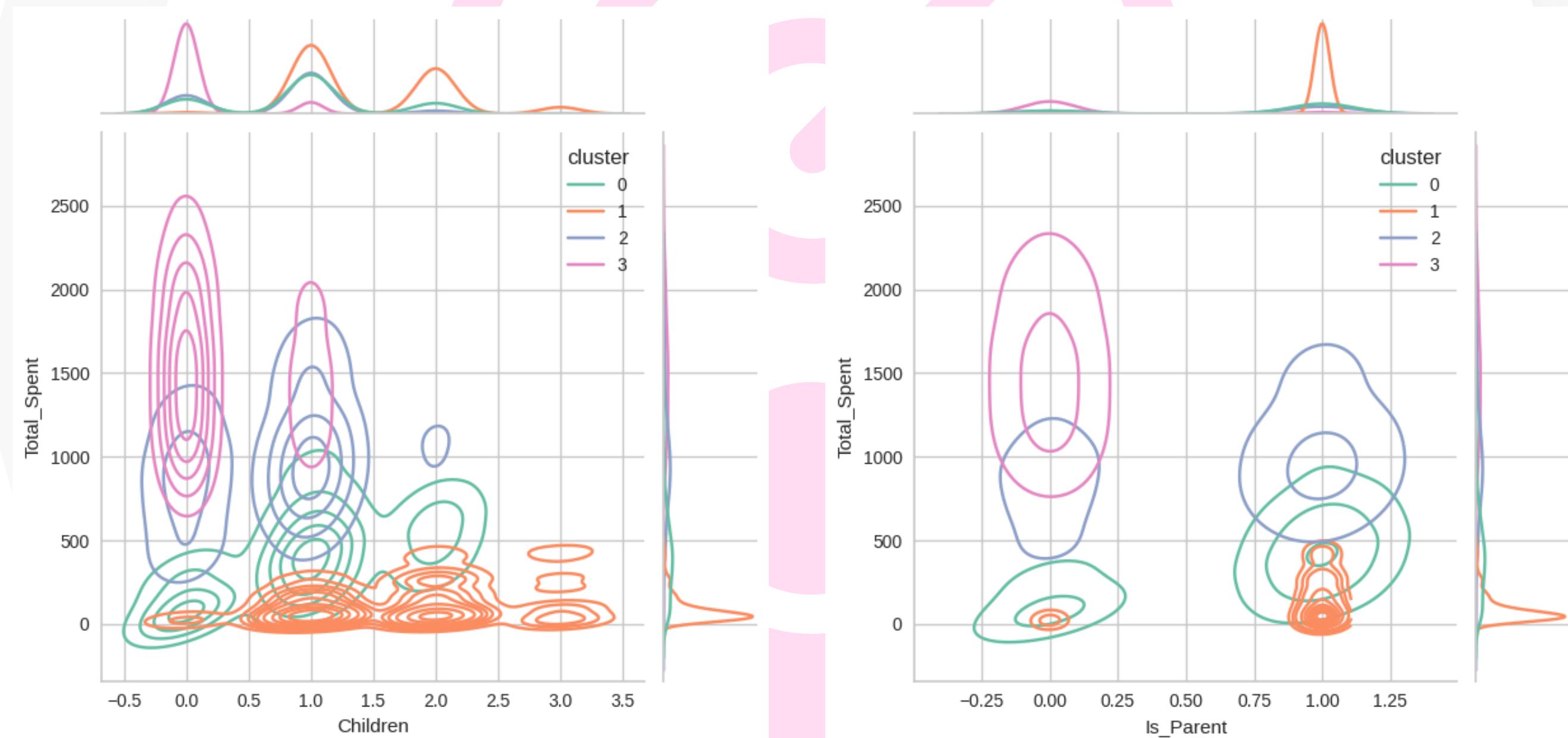
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

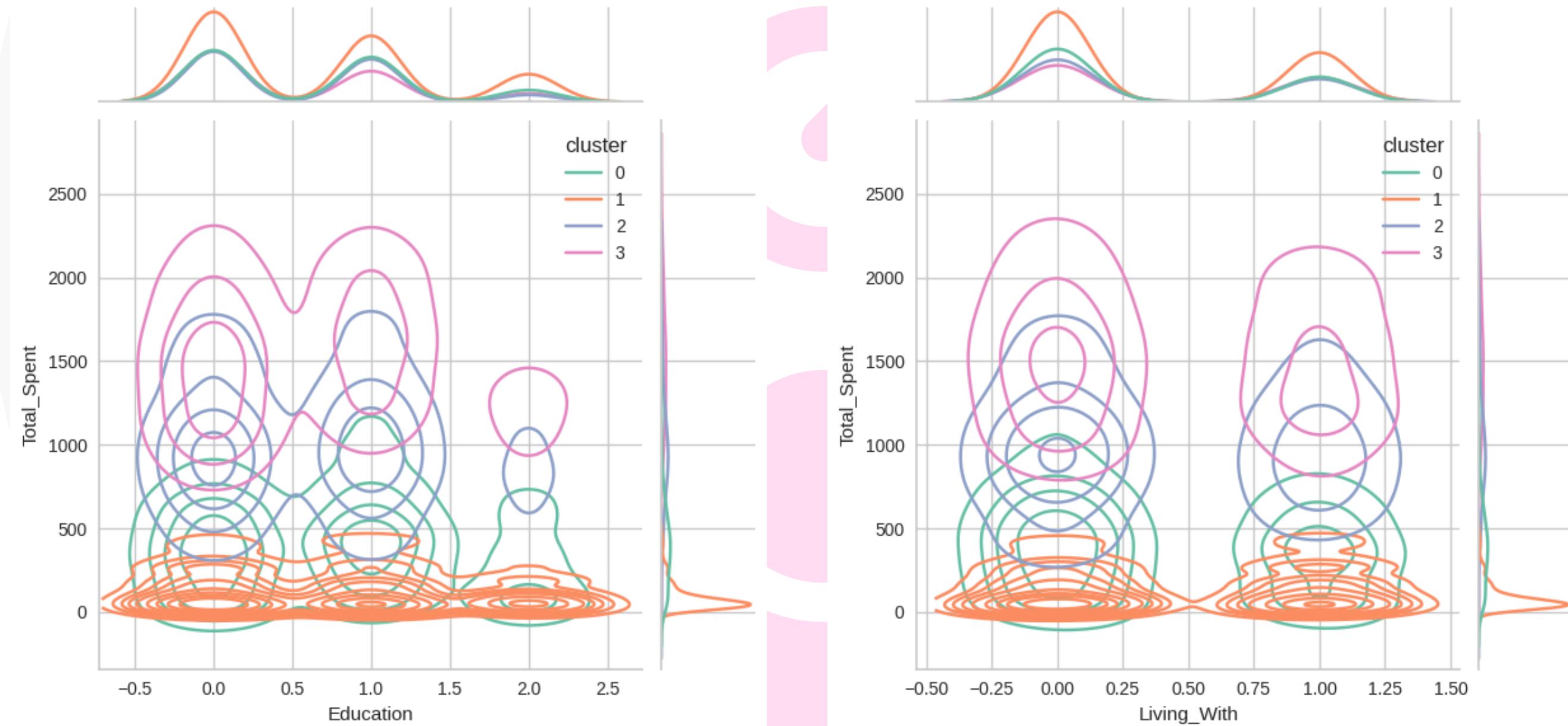
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

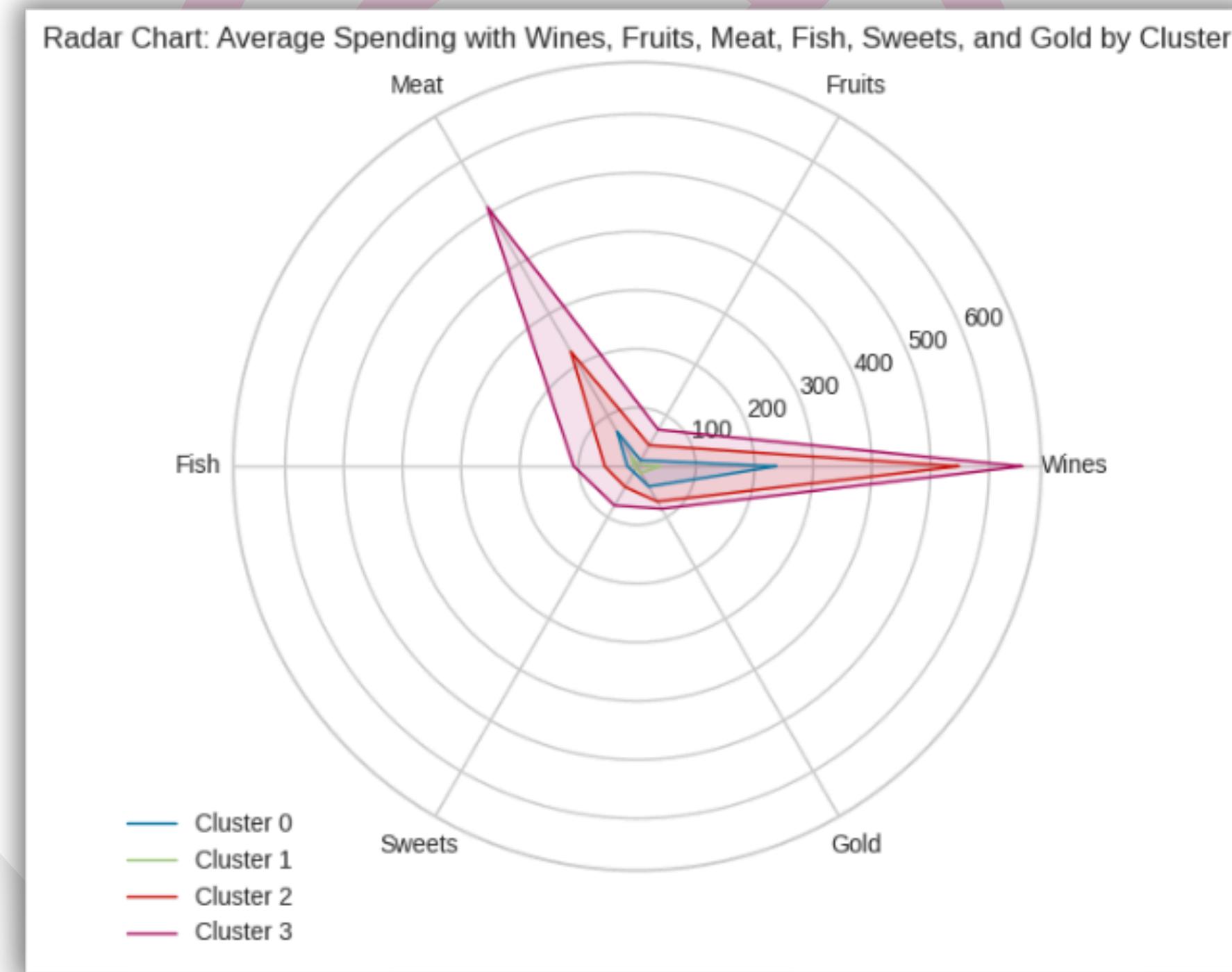
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

7. Xem xét các thói quen mua thực phẩm của các tập khách hàng.





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 1 ($K = 0$):

- Nhóm khách hàng chủ yếu là cặp bố mẹ
- Có cao nhất là 3 thành viên trong gia đình
- Nhóm khách hàng thường có 1 con nhỏ
- Độ tuổi nằm ở độ tuổi trưởng thành (30 đến 40 tuổi)
- Có độ tuổi tập trung từ 40 đến 60 tuổi
- Họ mua hàng rất ít





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 2 ($K = 1$):

- Nhóm khách hàng là bố hoặc mẹ
- Có cao nhất là 5 thành viên trong gia đình và thấp nhất là 2 thành viên
- Nhóm khách hàng có con ở độ tuổi vị thành niên
- Đa số khách hàng đều tập trung ở độ tuổi từ 40 đến 80 tuổi
- Là nhóm khách hàng già có thu nhập thấp
- Họ đa số mua nhiều sản phẩm rượu
-





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 3 ($K = 2$):

- Là những bố mẹ đơn thân
- Có số thành viên trong gia đình cao nhất là 4 người và thấp nhất là 2
- Hầu hết đều có con ở độ tuổi vị thành niên
- Đa số khách hàng mua nhiều sản phẩm rượu





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 4 ($K = 3$):

- Nhóm khách hàng chưa có con
- Có cao nhất là chỉ 2 thành viên trong gia đình
- Nhưng trong nhóm này đa số là cặp đôi yêu nhau
- Trải rộng ở mọi độ tuổi
- Và nhóm khách hàng này có thu nhập cao
- Họ thường mua thiên về cá thịt và rượu





THUẬT TOÁN

Hierarchical Clustering



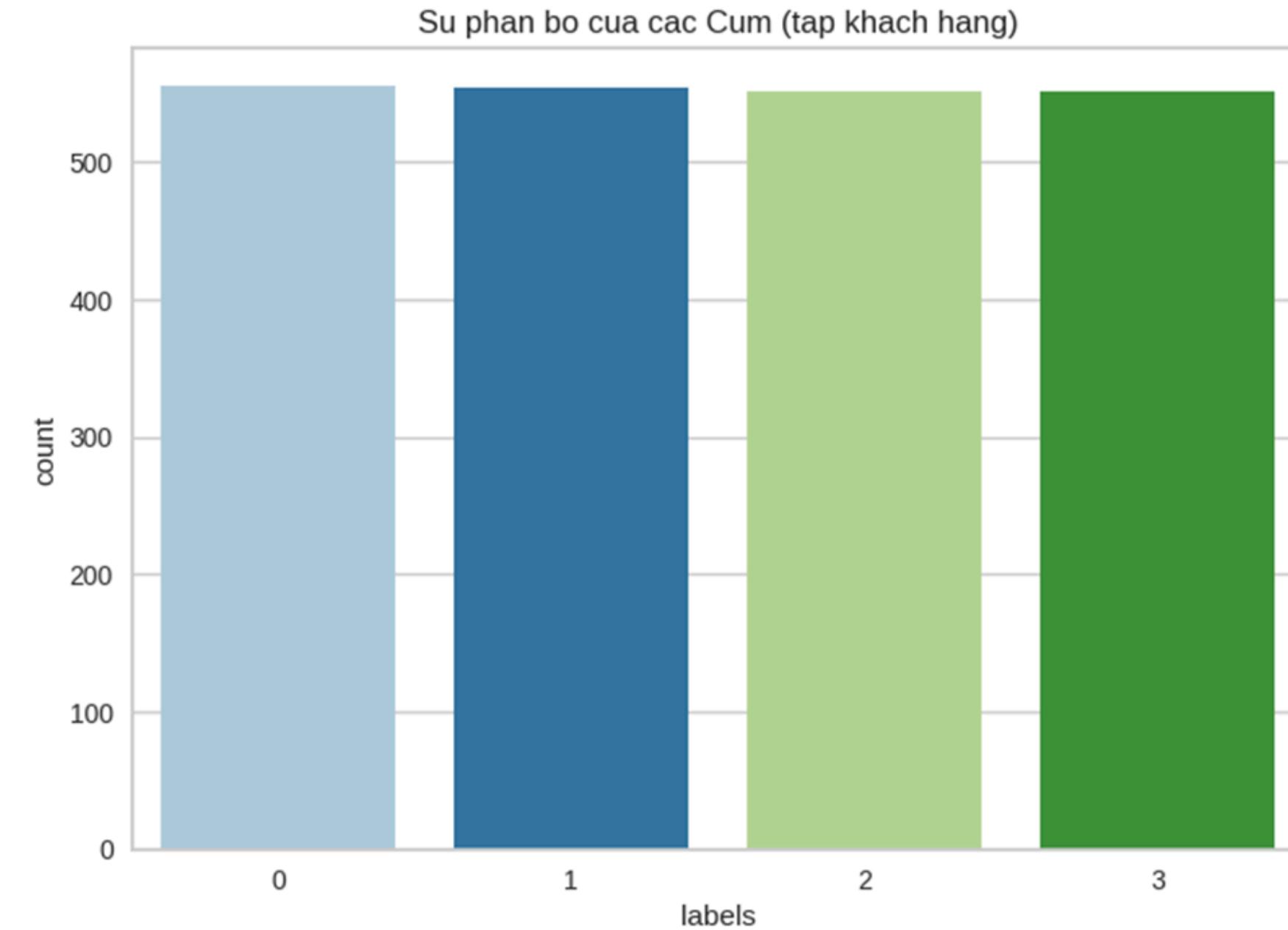
THUẬT TOÁN

Divise Hierarchical Clustering



IV. Kết quả đạt được và kết luận

1. Xem sự phân bố phân tử ở các cụm với nhau

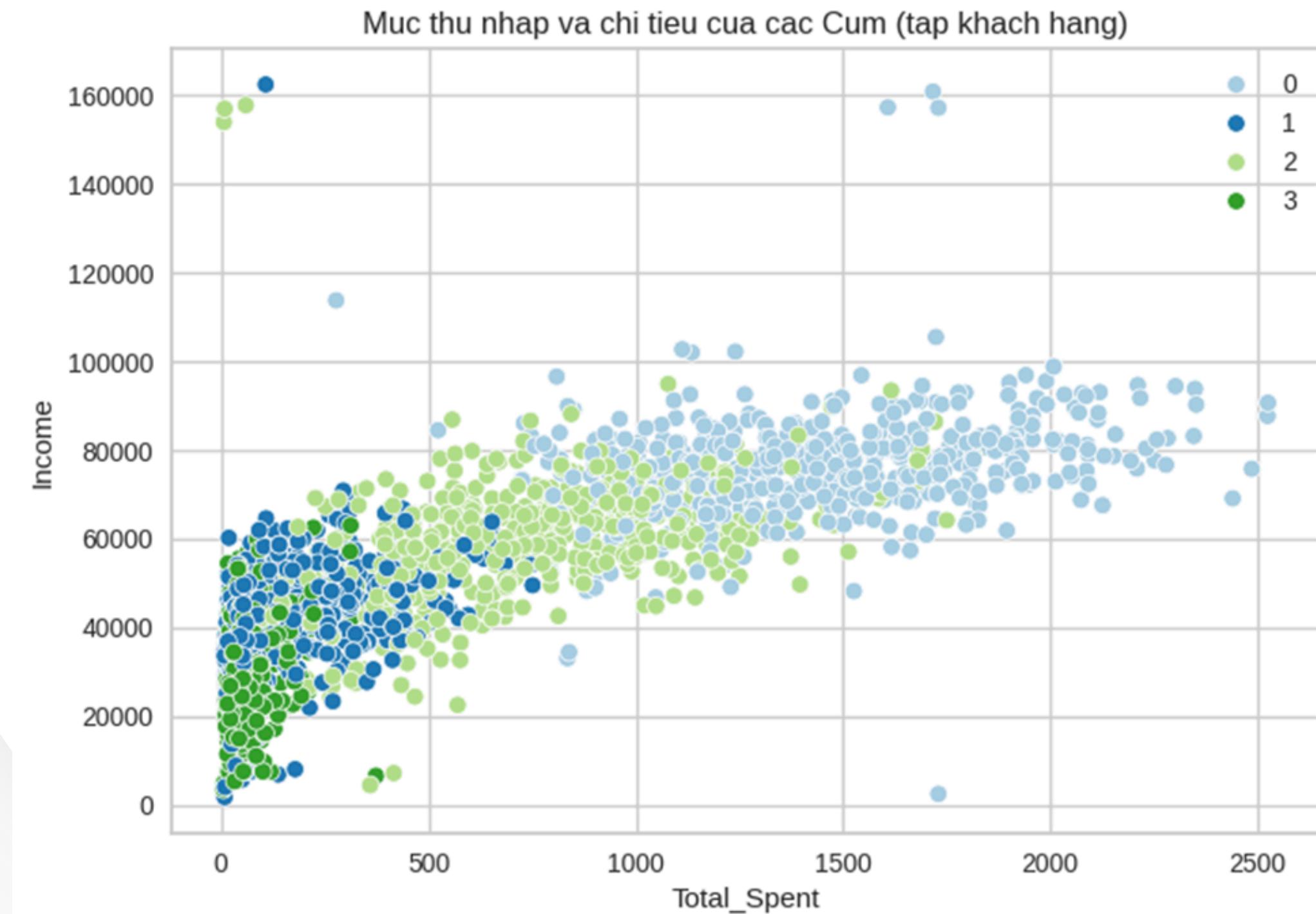


Nhận xét: kết quả phân cụm đều nhau. Một phần lý do là với cách hoạt động của thuật toán Divisive Hierarchical Clustering.



IV. Kết quả đạt được và kết luận

2. Xem xét sự phân bố thu nhập và chỉ tiêu ở các cụm với nhau





IV. Kết quả đạt được và kết luận

2. Xem xét sự phân bố thu nhập và chi tiêu ở các cụm với nhau

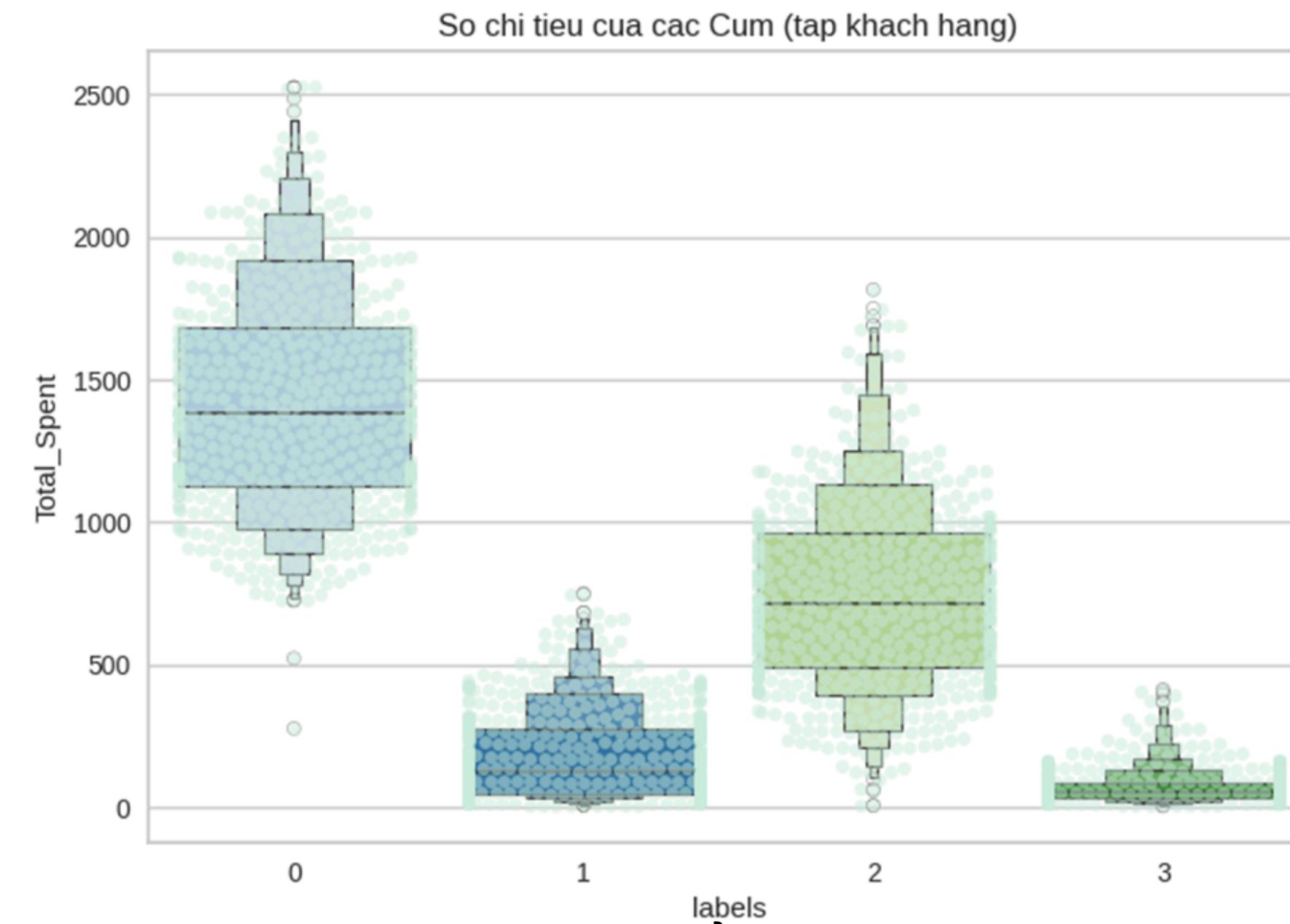
Nhận xét:

- Nhóm 1 (Cụm 0): Nhóm khách hàng có thu nhập và chi tiêu cao
- Nhóm 2 (Cụm 1): Nhóm khách hàng có thu nhập trung bình và chi tiêu thấp
- Nhóm 3 (Cụm 2): Nhóm khách hàng có thu nhập trung bình cao và chi tiêu trung bình
- Nhóm 4 (Cụm 3): Nhóm khách hàng có thu nhập trung bình thấp và chi tiêu thấp



IV. Kết quả đạt được và kết luận

3. Xem xét sự phân bố chi tiết các cụm theo các sản phẩm khác nhau trong dữ liệu

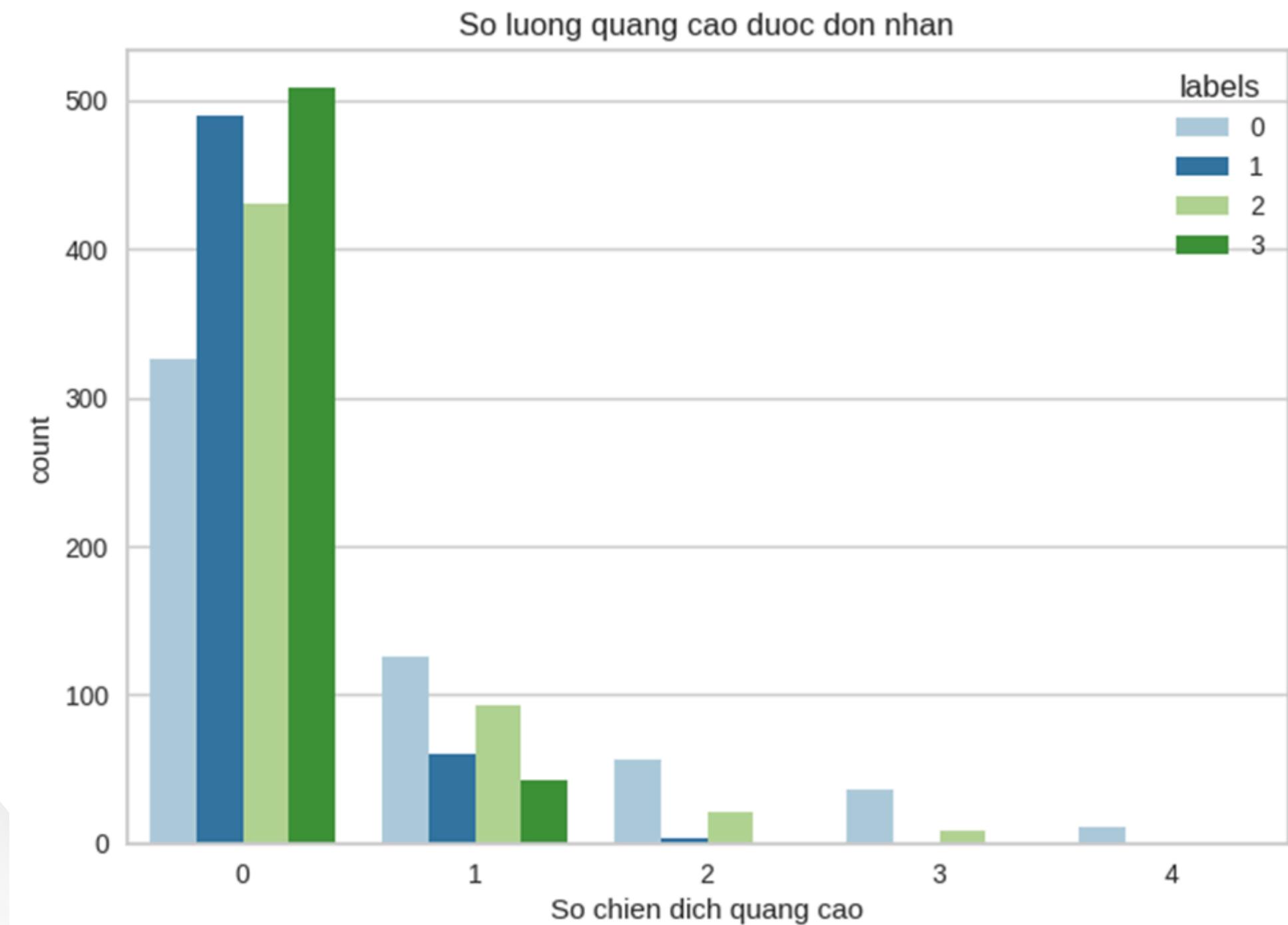


Nhận xét: Theo trực quan hóa dữ liệu. Có thể thấy rất rõ nhóm khách hàng thuộc cụm 0 là nhóm khách hàng lớn nhất. Theo sau đó là nhóm khách hàng thuộc cụm 2



IV. Kết quả đạt được và kết luận

4. Xem xét sự phân bố số lượng các chiến dịch quảng cáo và số ủng hộ





IV. Kết quả đạt được và kết luận

4. Xem xét sự phân bố số lượng các chiến dịch quảng cáo và số ủng hộ

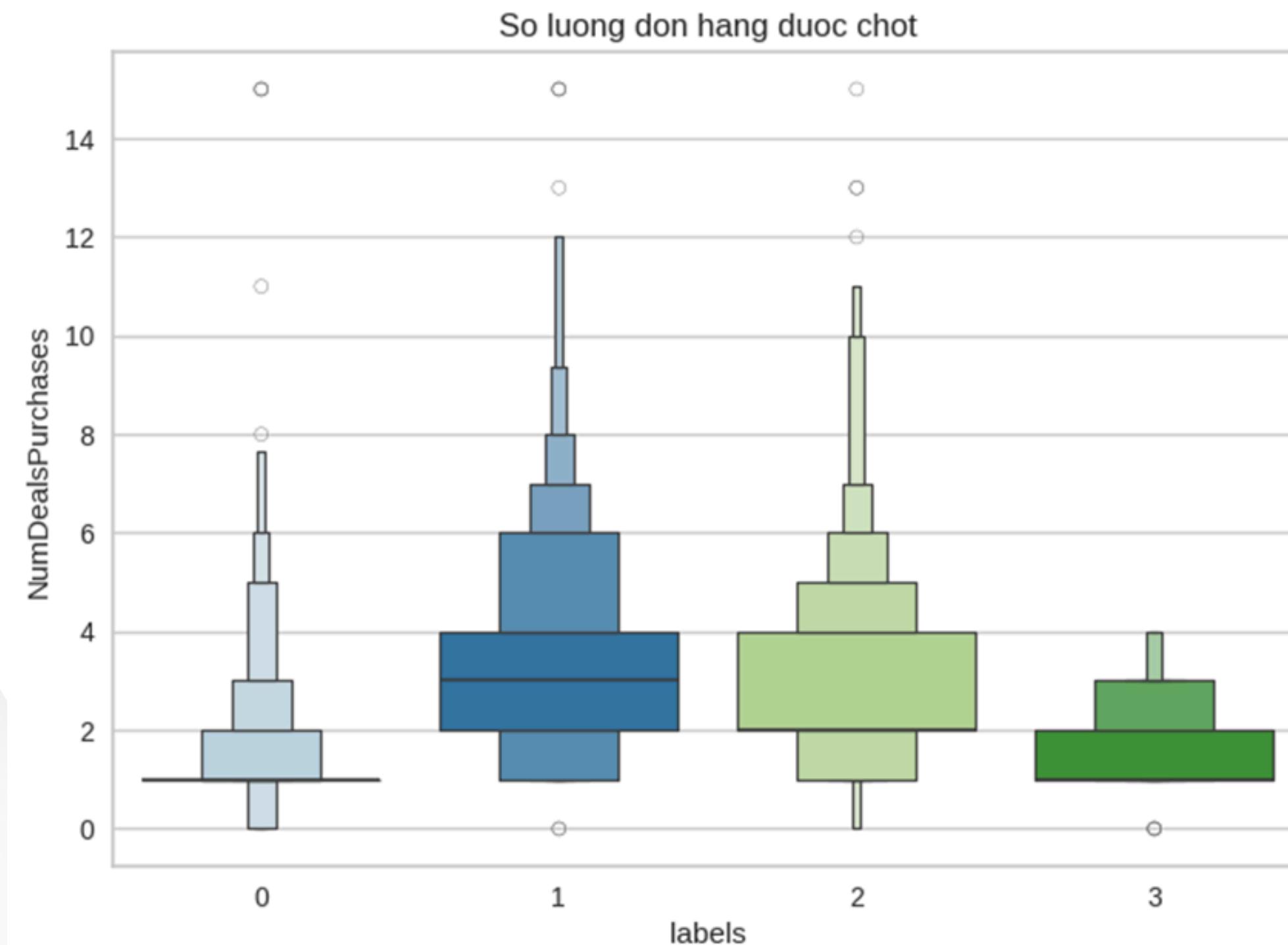
Nhận xét:

- Cho đến nay vẫn chưa có Cụm áp đảo nào dựa trên các chiến dịch.
- Nhìn chung rất ít người tham gia. Hơn nữa, không có Cụm nào rải đều cả 4 chiến dịch..
- Có lẽ cần phải có các chiến dịch được nhắm mục tiêu tốt hơn và được lên kế hoạch tốt hơn để tăng doanh số bán hàng.



IV. Kết quả đạt được và kết luận

5. Xem xét số lượng chốt đơn hàng giảm giá ở các tập khách hàng





IV. Kết quả đạt được và kết luận

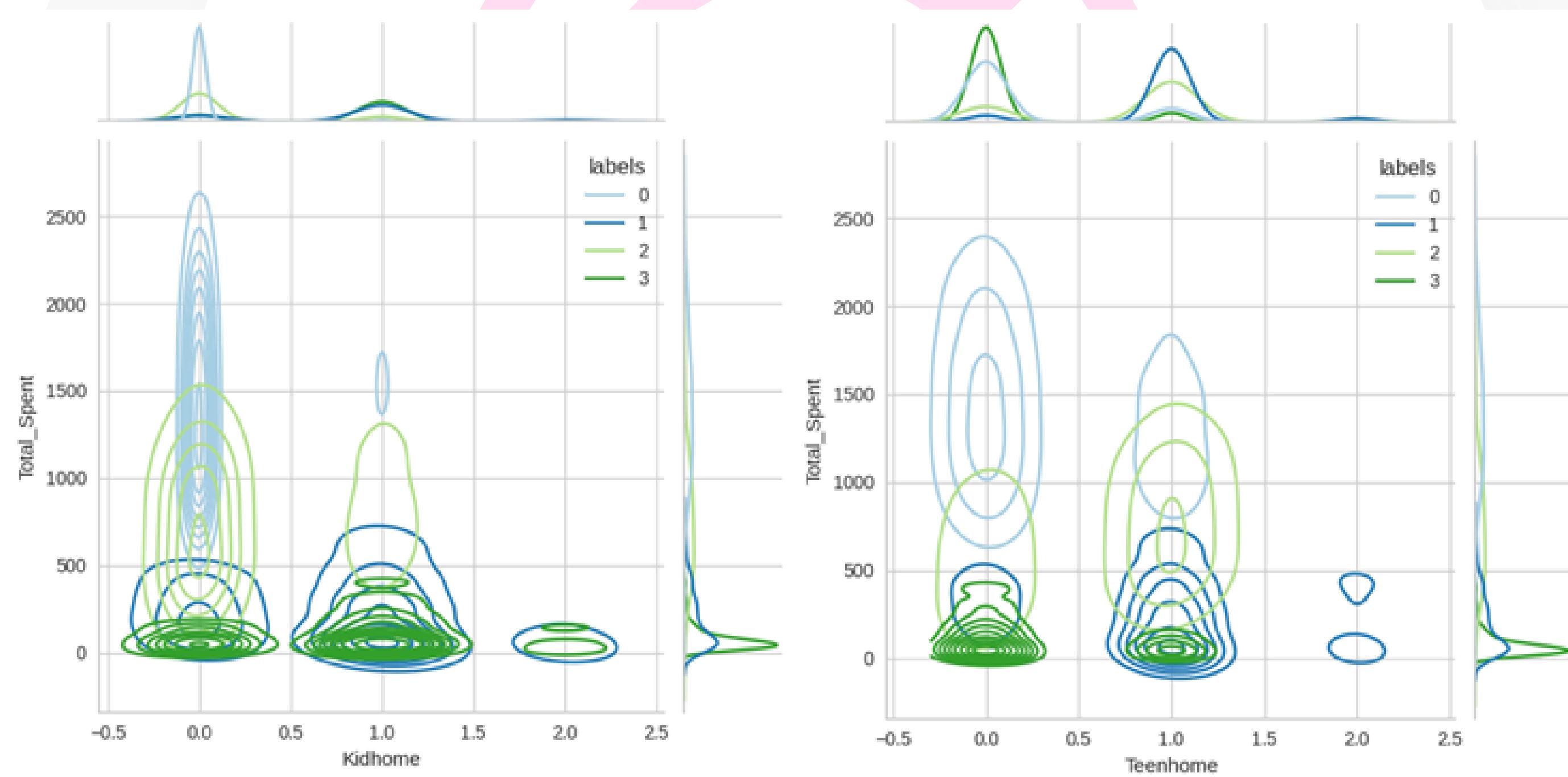
5. Xem xét số lượng chốt đơn hàng giảm giá ở các tập khách hàng

Nhận xét: Không giống như các chiến dịch, các hoạt động mua hàng giảm giá đều hoạt động tốt. Nó mang lại kết quả tốt nhất với cụm 1 và 2. Tuy nhiên, các khách hàng cụm 3 không quan tâm nhiều đến các hoạt động này.



IV. Kết quả đạt được và kết luận

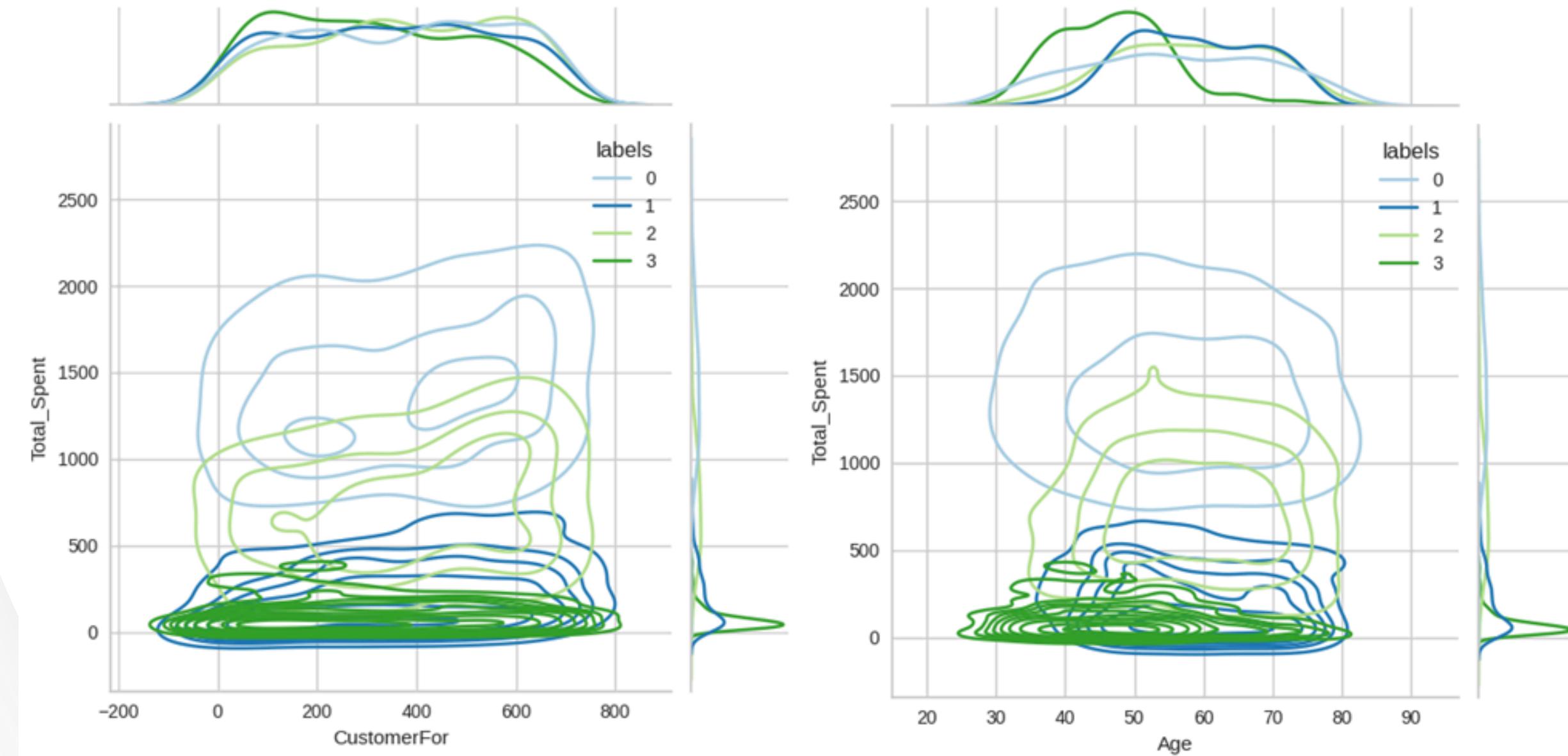
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

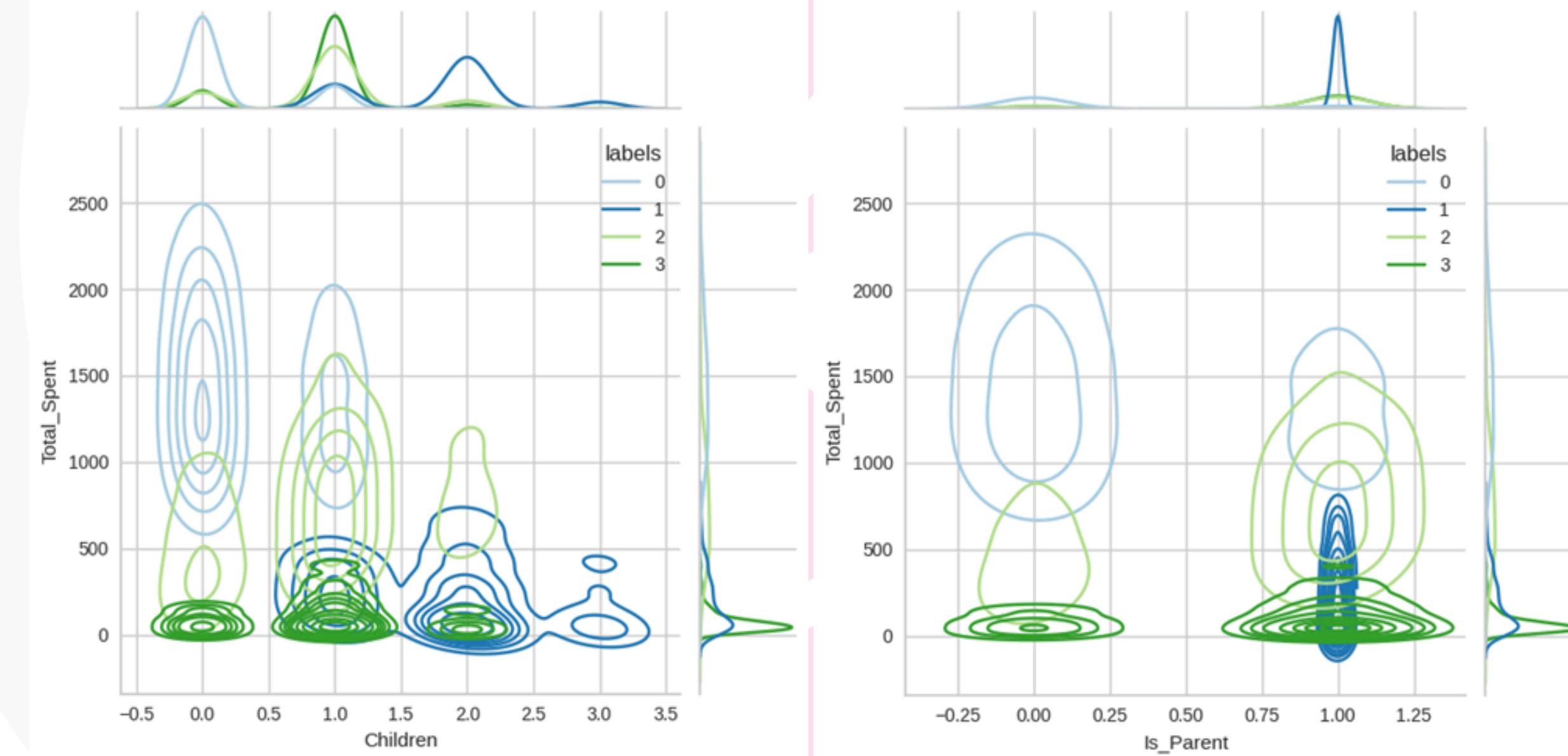
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

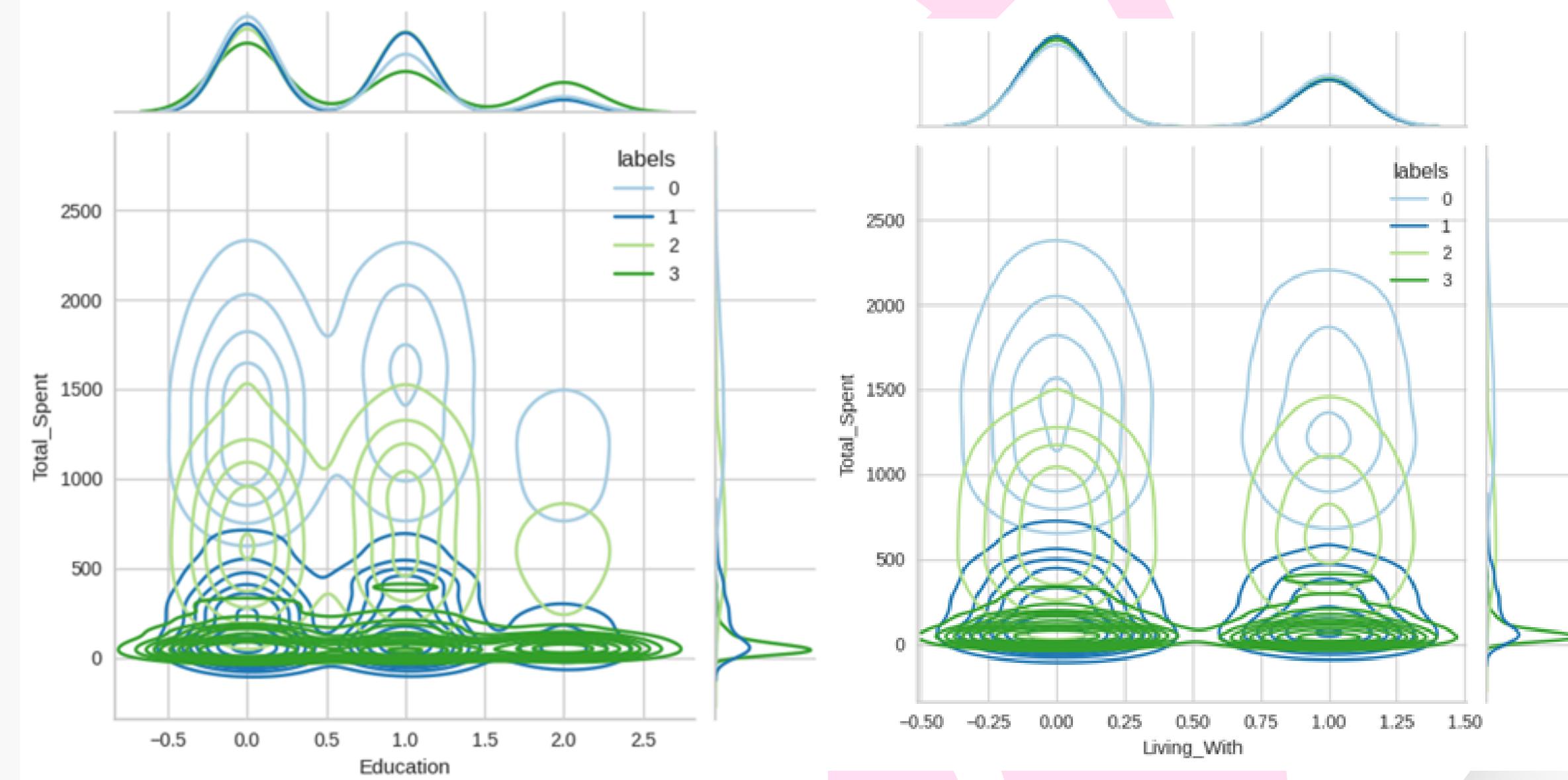
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

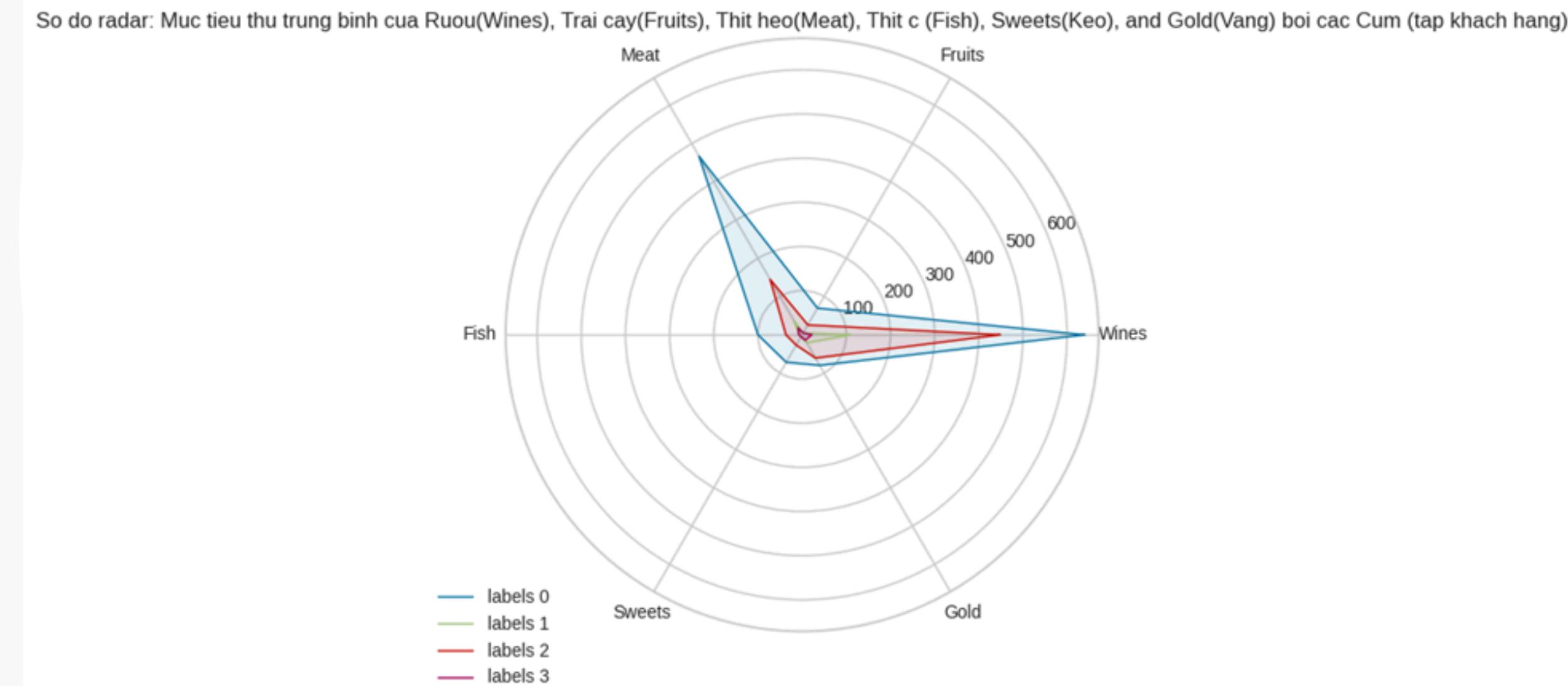
6. Xem xét các yếu tố nhân khẩu học





IV. Kết quả đạt được và kết luận

7. Xem xét các thói quen mua thực phẩm của các tập khách hàng.





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 1 ($K = 0$):

- Nhóm khách hàng chủ yếu là khách hàng chưa có con
- Có cao nhất là 3 thành viên trong gia đình
- Độ tuổi rải rác từ 30 đến 70
- Nhưng trong nhóm này đa số là cặp đôi yêu nhau
- Nhóm khách hàng thường có 1 con nhỏ
- Có cao nhất là 3 thành viên trong gia đình
- Họ mua hàng rất nhiều, chủ yếu thịt và rượu





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 2 ($K = 1$):

- Nhóm khách hàng là bố mẹ có con
- Có cao nhất là 5 thành viên trong gia đình và thấp nhất là 3 thành viên
- Nhóm khách hàng có con ở độ tuổi vị thành niên
- Đa số khách hàng đều tập trung ở độ tuổi từ 40 đến 80 tuổi
- Họ đa số mua rượu nhưng ít





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 3 ($K = 2$):

- Là những bố mẹ đơn thân
- Độ tuổi trải đều từ 35 đến 75 tuổi.
- Có số thành viên trong gia đình cao nhất là 4 người và thấp nhất là 2
- Hầu hết đều có con ở độ tuổi vị thành niên
- Đa số khách hàng mua nhiều sản phẩm rượu





IV. Kết quả đạt được và kết luận

Thông qua việc trực quan hóa, có thể thấy được 4 nhóm khách hàng có những đặc trưng khác nhau

Về cụm khách hàng 4 ($K = 3$):

- Chú yếu là bố mẹ với con nhỏ
- Học vấn trải đều
- Có cao nhất là chỉ 3-4 thành viên trong gia đình
- Trải rộng ở mọi độ tuổi
- Và nhóm khách hàng này có thu nhập thấp
- Mua ít sản phẩm





So sánh & Đánh giá





Bảng đánh giá

	K-means	AHC	DHC
Silhouette Coefficient	-0.4810014358639396	0.1695101794331076	0.30985129394566485
calinski harabasz	2527.5591901852004	0.9461160885532321	1640.7549164975528



Cảm ơn thầy đã
lắng nghe!

