

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN KHAI PHÁ DỮ LIỆU



Môn học: Khai phá dữ liệu

Mã môn học: DAMI330484_22_2_01

KHAI PHÁ DỮ LIỆU THÔNG TIN GIAO DỊCH CỦA KHÁCH HÀNG

GVHD: Ths. Nguyễn Văn Thành

Sinh viên thực hiện: Nhóm 19

Nguyễn Minh Tiến	20133093
Huỳnh Nguyễn Tín	20133094
Bùi Lê Hải Triều	20133101
Đỗ Hoàng Thịnh	20133122

TP.HCM, tháng 05 năm 2023

Lời cảm ơn

Nhóm chúng em xin gửi lời tri ân đến Khoa Công Nghệ Thông Tin – Trường Đại Học Sư Phạm Kỹ Thuật Thành Phố Hồ Chí Minh vì đã cho nhóm chúng em cơ hội học tập, nắm vững kiến thức nền tảng và thực hiện đề tài này.

Nhờ có sự hướng dẫn tận tâm, giảng dạy đầy đủ kiến thức của thầy Nguyễn Văn Thành, chúng em đã được tiếp cận những kiến thức quan trọng về môn Khai phá dữ liệu.

Qua đó chúng em biết cách sử dụng các thuật toán khai phá dữ liệu để hỗ trợ cho việc học tập và tìm hiểu chuyên sâu. Tuy nhiên kiến thức là bao la và với khả năng có hạn chúng em đã cố gắng hết sức để hoàn thành đề tài một cách tốt nhất. Chúng em mong muốn nhận được sự góp ý của thầy để từ đó chúng em có thể rút ra được bài học kinh nghiệm và hoàn thiện và cải thiện sản phẩm của mình một cách tốt nhất có thể. Chúng em xin chân thành cảm ơn!

- **Bảng phân công công việc**

Nhóm 19				
	Nguyễn Tín	Minh Tiến	Hải Triều	Hoàng Thịnh
Tiền xử lý dữ liệu	x	x	x	x
FpGrowth		x	x	x
Apriori	x	x		x
Kmeans			x	
Classicfication	x			
Viết báo cáo	x	x	x	x

Mục lục

PHẦN MỞ ĐẦU.....	1
PHẦN NỘI DUNG.....	2
1. Clustering	2
1.1. Giới thiệu thuật toán	2
1.2. K-Means Clustering.....	2
1.3. Thực hiện mô hình K-means clustering bằng python.....	4
2. Association rules	10
2.1. Tiền xử lý dữ liệu	11
2.2 Sử dụng thuật toán FpGrowth	13
2.3 Sử dụng thuật toán Apriori	15
3. Association rules trên SSAS.....	16
3.1. Chuẩn bị dữ liệu.....	16
3.2. Mining Structure on SSAS.....	21
5. Classification	35
5.1. Decision tree	35
5.2. Selecting features	35
5.3. Transforming data.....	37
5.4. Splitting train and test data	37
5.5. Build model.....	38
5.6. Improving accuracy	38

PHẦN MỞ ĐẦU

- **Tính cấp thiết của đề tài**

Việc phân tích thông tin giao dịch của các khách hàng là một hoạt động quan trọng để nâng cao hiệu quả kinh doanh và tiếp thị. Thông qua việc phân tích thông tin giao dịch, các chủ đầu tư và quản lý trung tâm mua sắm có thể hiểu rõ hơn về hành vi, nhu cầu và mong muốn của khách hàng, từ đó đưa ra các chiến lược phù hợp để thu hút và giữ chân khách hàng. Ngoài ra, việc phân tích thông tin giao dịch cũng giúp các trung tâm mua sắm nắm bắt được xu hướng thị trường, cạnh tranh và đổi mới sản phẩm, dịch vụ để tạo ra sự khác biệt và giá trị gia tăng cho khách hàng. Việc phân tích thông tin giao dịch của khách hàng có thể được thực hiện bằng cách sử dụng các công cụ và phương pháp thống kê, khai phá dữ liệu, học máy và trí tuệ nhân tạo để xử lý và phân tích các dữ liệu thu thập được từ các kênh giao dịch như thẻ thành viên, hóa đơn, phiếu khảo sát, website, ứng dụng di động, mạng xã hội... Từ đó, có thể rút ra các thông tin hữu ích về đặc điểm cá nhân, sở thích, tần suất mua hàng, giá trị giao dịch, mức độ hài lòng và trung thành của khách hàng với các trung tâm mua sắm.

Trong đề tài này, chúng em sẽ sử dụng các thuật toán phân cụm, phân loại và kết hợp, ngoài ra nhóm cũng sử dụng thêm công cụ SSAS để phục vụ cho mục đích phân tích và làm rõ hơn về một tập dữ liệu lớn.

- **Đối tượng nghiên cứu**

Nhóm sử dụng tập dữ liệu chứa thông tin giao dịch của khách hàng từ 10 trung tâm mua sắm lớn tại đất nước Istanbul, từ năm 2021 đến thời điểm hiện tại năm 2023. Ngoài thông tin giao dịch, tập dữ liệu cũng cung cấp thông tin về độ tuổi, giới tính, phù hợp với nghiệp vụ khai phá.

PHẦN NỘI DUNG

1. Clustering

1.1. Giới thiệu thuật toán

Thuật toán clustering là một phương pháp phân tích dữ liệu không giám sát, nghĩa là không cần nhãn của các điểm dữ liệu. Mục tiêu của thuật toán clustering là phân chia các điểm dữ liệu thành các nhóm (cluster) sao cho các điểm trong cùng một nhóm có tính tương đồng cao, còn các điểm ở các nhóm khác nhau có tính tương đồng thấp. Có nhiều loại thuật toán clustering khác nhau, ví dụ như k-means, hierarchical clustering, DBSCAN, và spectral clustering. Các thuật toán này có cách tìm kiếm và xác định các nhóm khác nhau tùy theo các tiêu chí và đặc trưng của dữ liệu.

Trong số các thuật toán clustering khác nhau, nhóm em quyết định chọn k-means làm thuật toán để phân cụm các khách hàng trong tập dữ liệu.

1.2. K-Means Clustering

K-means clustering là một thuật toán phân cụm dữ liệu không giám sát, nghĩa là không cần nhãn cho các điểm dữ liệu. Thuật toán này nhằm mục đích chia các điểm dữ liệu thành k nhóm sao cho mỗi nhóm có các điểm gần nhau nhất về mặt không gian. K-means clustering có thể được coi là một phương pháp tối ưu hóa, vì nó cố gắng tìm ra k trung tâm cụm (cluster center) tối ưu để giảm thiểu hàm mất mát (loss function), là tổng bình phương khoảng cách từ mỗi điểm đến trung tâm cụm gần nhất.

Thuật toán K-means clustering có các bước thực hiện như sau:

- **Bước 1:** Chọn ngẫu nhiên k điểm làm trung tâm cụm ban đầu.
- **Bước 2:** Gán mỗi điểm dữ liệu vào cụm có trung tâm gần nó nhất theo khoảng cách Euclid.

- **Bước 3:** Cập nhật lại trung tâm cụm bằng cách tính trung bình các điểm trong cùng một cụm.
- **Bước 4:** Lặp lại bước 2 và 3 cho đến khi trung tâm cụm không thay đổi hoặc đạt được số vòng lặp tối đa.

K-means clustering có nhiều ứng dụng trong các lĩnh vực như thống kê, máy học, khai phá dữ liệu, phân tích hình ảnh, nén dữ liệu, phát hiện ngoại lai và phân loại văn bản.

So sánh với các thuật toán khác:

K-means clustering là một thuật toán phân cụm dựa trên trung tâm (centroid-based), tức là sử dụng các vector đặc trưng để biểu diễn các cụm. Các thuật toán khác trong loại này là k-medoids, k-medians, k-modes và k-prototypes. Điểm khác biệt chính giữa chúng là cách tính trung tâm cụm và khoảng cách giữa các điểm.

Ngoài ra, có các thuật toán phân cụm dựa trên phân phối (distribution-based), ví dụ như Gaussian Mixture Model (GMM), Expectation-Maximization (EM) và Dirichlet Mixture Model (DMM). Các thuật toán này giả định rằng các điểm dữ liệu được sinh ra từ một số phân phối xác suất đã biết và cố gắng ước lượng các tham số của phân phối đó.

Một loại thuật toán phân cụm khác là dựa trên mật độ (density-based), ví dụ như DBSCAN, DENCLUE và OPTICS. Các thuật toán này xác định các cụm là các vùng có mật độ cao hơn so với các vùng xung quanh và có thể xử lý được các dữ liệu có hình dạng phi tuyến và nhiễu.

1.3. Thực hiện mô hình K-means clustering bằng python

Import các thư viện cần thiết

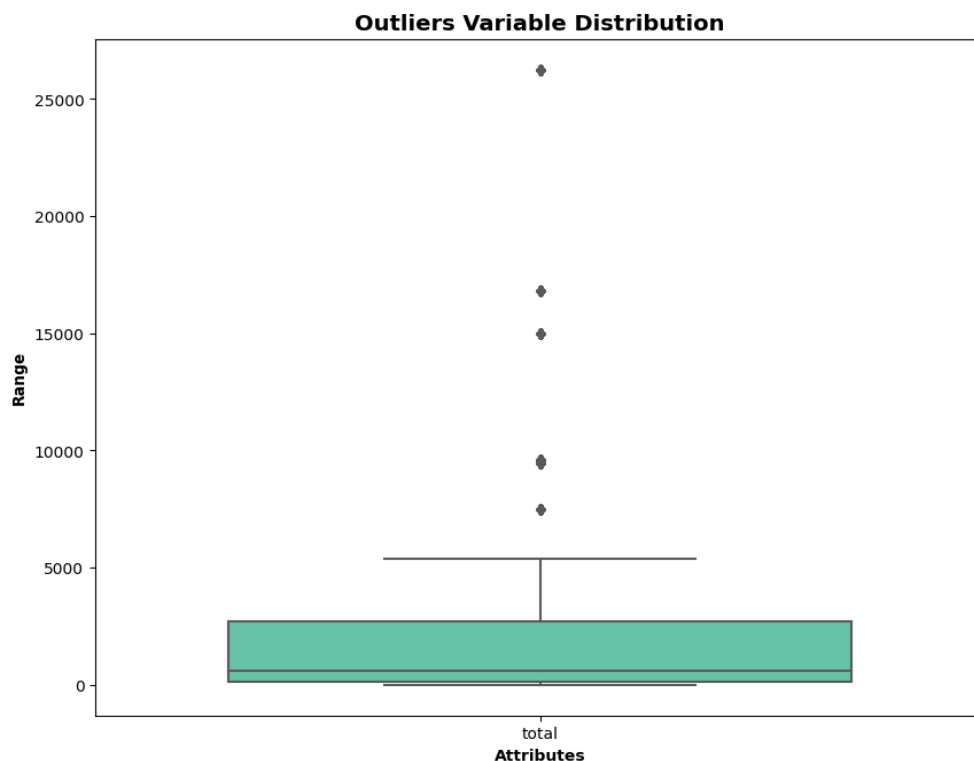
```
from sklearn.preprocessing import MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import numpy as np
5]
```

Dưới đây sẽ thực hiện việc chuẩn bị data trước khi đưa vào xây dựng mô hình.

Tạo một dataframe gồm các cột 'age', 'total' là các thuộc tính dùng để phân loại khách hàng.

```
df_new=Kmeans_df[['age','total']]
```

Biểu đồ boxplot thể hiện phân bố giá trị của cột total



Để cải thiện hiệu quả của thuật toán ta thực hiện việc loại bỏ các outliers của cột total.

```
#loại bỏ outliers ở cột total
Q1 = df_new.total.quantile(0.05) #xác định tứ phân vị đầu
Q3 = df_new.total.quantile(0.95) #xác định tứ phân vị thứ 3
IQR = Q3 - Q1 #tính IQR
df_new = df_new[(df_new.total >= Q1 - 1.5*IQR) & (df_new.total <= Q3 + 1.5*IQR)]
#loại bỏ các giá trị ở cột total mà bé hơn giá trị biên dưới và lớn hơn biên trên
```

Thực hiện scaling data.

```
# Select the age and amount columns
df_scaled = df_new[['age', 'total']]

# Scale the age and amount columns using min-max scaling
scaler = MinMaxScaler()
df_scaled = scaler.fit_transform(df_scaled)
df_scaled = pd.DataFrame(df_scaled, columns=['age', 'total'])

df_scaled = pd.DataFrame(df_scaled)
df_scaled.columns = ['age', 'total']
df_scaled.head()
```

	age	total
0	0.196078	0.446375
1	0.058824	0.321308
2	0.039216	0.017556
3	0.941176	0.893077
4	0.686275	0.014122

Hoàn thành việc chuẩn bị data cho việc xây dựng mô hình.

Tiếp theo, để đạt được mô hình K-means clustering tối ưu, ta cần tìm ra số K cụm tối ưu bằng 2 phương pháp

Thứ nhất là **Elbow Curve**.

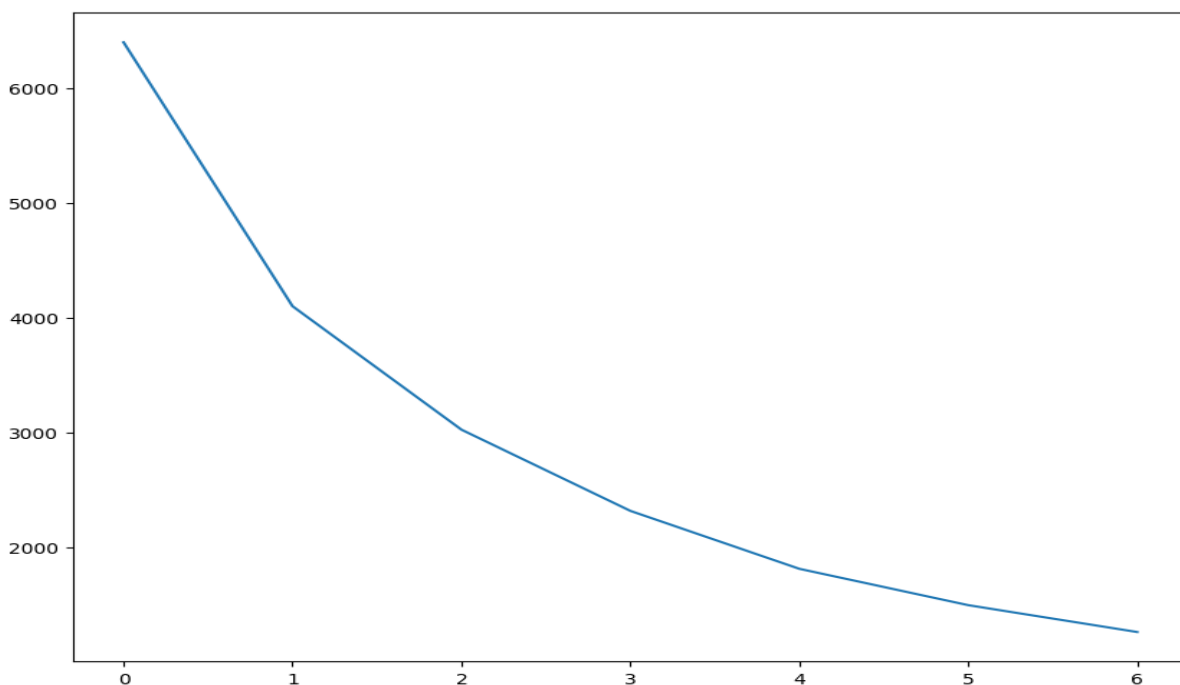
Elbow curve là một biểu đồ thể hiện mối quan hệ giữa số lượng các nhóm dữ liệu được phân cụm và độ sai lệch trung bình của các điểm dữ liệu so với trung tâm nhóm. Elbow curve có thể giúp chọn số lượng nhóm phù hợp cho phân tích phân cụm. Điểm giao nhau giữa hai đoạn thẳng trên biểu đồ được gọi là elbow point, nơi mà độ sai lệch trung bình bắt đầu giảm chậm hơn khi tăng số lượng nhóm.

Code python thực hiện Elbow Curve

```
ssd = [] # khởi tạo danh sách trống để lưu trữ các tổng bình phương khoảng cách (sum of squared distance) cho mỗi giá trị k khác nhau
range_n_clusters = [2, 3, 4, 5, 6, 7, 8] # khởi tạo các giá trị cho K cluster
for num_clusters in range_n_clusters: # vòng lặp với mỗi giá trị k từ 2 đến 8
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50)
    kmeans.fit(df_scaled)

    ssd.append(kmeans.inertia_) # hàm inertia tính tổng bình phương khoảng cách của mô hình vừa train

# Biểu diễn các giá trị ssd theo Elbow Curve
plt.plot(ssd)
```



Biểu đồ trên cho thấy tại $k=3$ hoặc $k=4$ đồ thị bắt đầu giảm dần đều cho thấy độ sai lệch trung bình của các điểm dữ liệu so với trung tâm nhóm không chênh lệch nhiều.

Để chọn ra số k cụm đúng thì ta sẽ thực hiện thêm một phương pháp thứ hai đó là **Silhouette Score**

Silhouette Score là một phương pháp đo lường độ tương đồng của các điểm dữ liệu trong cùng một nhóm so với các nhóm khác. Silhouette Score có giá trị từ -1 đến 1, trong đó giá trị cao hơn thể hiện sự phân cụm tốt hơn. Để tính Silhouette Score cho một điểm dữ liệu, ta cần xác định khoảng cách trung bình của nó đến các điểm khác trong cùng nhóm (a) và khoảng cách trung bình của nó đến các điểm gần nhất trong nhóm khác (b). Sau đó, Silhouette Score được tính bằng công thức: $\text{Silhouette Score} = (b - a) / \max(a, b)$.

Code python thực hiện tìm Silhouette Score

```
# Silhouette analysis
range_n_clusters = [2, 3, 4, 5, 6]
#tạo vòng lặp thực hiện tính điểm trung bình silhouette của từng lần thực thi thuật toán Kmeans cùng số K cluster tương ứng
for num_clusters in range_n_clusters:

    # khởi tạo kmeans
    kmeans = KMeans(n_clusters=num_clusters, max_iter=30)
    kmeans.fit(df_scaled)

    cluster_labels = kmeans.labels_ #gán mảng giá trị cluster tương ứng

    # silhouette score
    silhouette_avg = silhouette_score(df_scaled, cluster_labels) #thực hiện tính điểm trung bình Silhouette trong dataframe df_scaled với số cụm được truyền vào
    print("For n_clusters={0}, the silhouette score is {1}".format(num_clusters, silhouette_avg))

For n_clusters=2, the silhouette score is 0.4721108234325241
For n_clusters=3, the silhouette score is 0.5040912376013137
For n_clusters=4, the silhouette score is 0.43623872125719537
For n_clusters=5, the silhouette score is 0.4619227076843678
For n_clusters=6, the silhouette score is 0.48378754912942773
```

Để thấy tại $n_clusters=3$ đạt giá trị silhouette score cao nhất cho thấy sự phân cụm tốt nhất

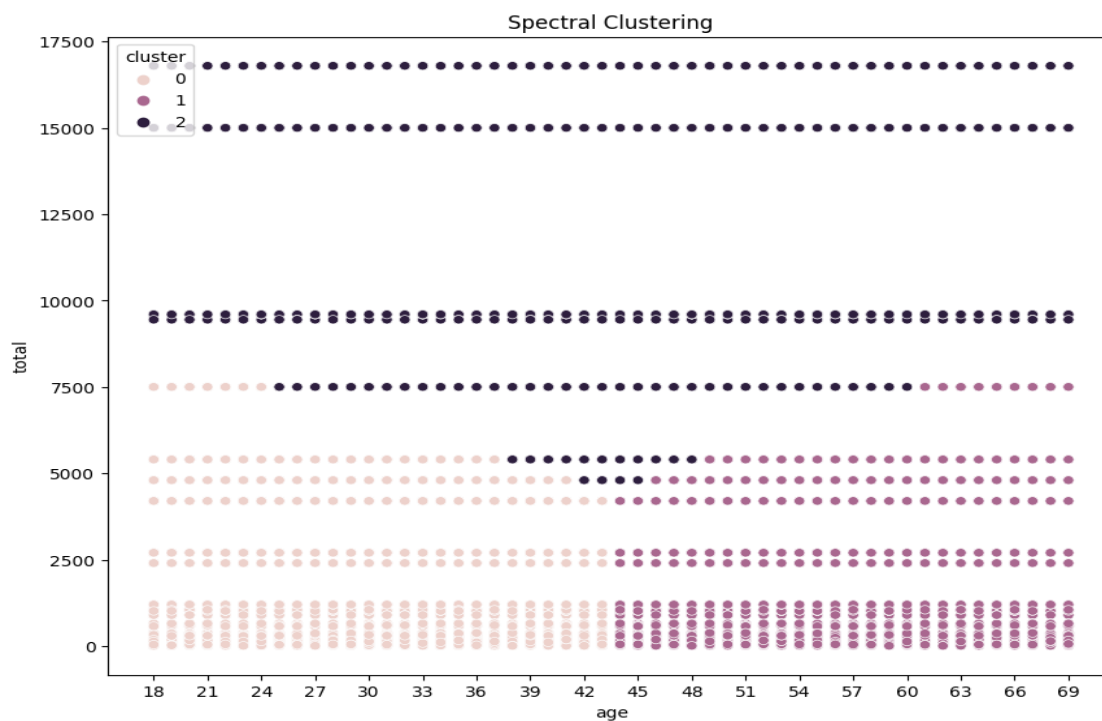
Suy ra ta chọn $k=3$ là số cụm dùng để xây dựng mô hình

```
kmeans = KMeans(n_clusters=3, max_iter=50)
kmeans.fit(df_scaled)

# kết hợp tên cụm vào dataframe df_new, kmeans.labels_
df_new['cluster'] = kmeans.labels_
df_new.head()
```

	age	total	cluster
0	28	7502.00	2
1	21	5401.53	0
2	20	300.08	0
3	66	15004.25	2
4	53	242.40	1

Trực quan hóa dữ liệu sau khi phân cụm theo age và total



Kết luận: Phân được 3 cụm

- **Cụm 1** là nhóm khách hàng trẻ tuổi, có thu nhập thấp hoặc tiết kiệm, không quan tâm nhiều đến mua sắm. Họ có thể là những người mới ra trường, sinh viên hoặc những người có lối sống giản dị.
- **Cụm 2** là nhóm khách hàng trung niên hoặc cao tuổi, có thu nhập ổn định, quan tâm đến mua sắm nhưng không quá nhiều. Họ có thể là những người đã có gia đình, nghề nghiệp và sở thích riêng.
- **Cụm 3** là nhóm khách hàng đa dạng về độ tuổi, có thu nhập từ vừa đến cao, quan tâm nhiều đến mua sắm. Họ có thể là những người thích theo kịp xu hướng, thể hiện cá tính và sở hữu những sản phẩm chất lượng.

Vài phương án kích thích chi tiêu khách hàng theo từng cụm:

- **Đối với cụm 1:** có thể tạo ra những chương trình khuyến mãi, giảm giá, tặng quà hoặc điểm thưởng để kích thích họ mua sắm nhiều hơn. Ngoài ra, có thể tìm hiểu nhu cầu và sở thích của họ để đưa ra những sản phẩm phù hợp với túi tiền và gu thẩm mỹ của họ.
- **Đối với cụm 2:** có thể tăng cường chất lượng dịch vụ, chăm sóc khách hàng và bảo hành sản phẩm để tạo ra sự tin tưởng và trung thành của họ. Ngoài ra, có thể giới thiệu những sản phẩm mới, độc đáo hoặc cao cấp để thu hút sự chú ý và thỏa mãn nhu cầu của họ.
- **Đối với cụm 3:** có thể tận dụng các kênh truyền thông, quảng cáo và marketing để nâng cao nhận diện thương hiệu và tạo ra sự lan tỏa của sản phẩm. Ngoài ra, có thể tạo ra những trải nghiệm mua sắm đặc biệt, cá nhân hóa hoặc tương tác để tăng sự hài lòng và gắn bó của họ.

2. Association rules

Quy luật liên kết (association rules) là một khái niệm trong lĩnh vực dữ liệu khai thác (data mining) và phân tích dữ liệu (data analysis). Đây là một phương pháp phân tích mối quan hệ giữa các mục (items) trong một tập dữ liệu. Mục đích chính của quy luật liên kết là tìm ra các mô hình liên kết, hay các quy tắc, mà những quy tắc đó thường xuất hiện cùng nhau trong tập dữ liệu.

Quy luật liên kết được biểu diễn dưới dạng "Nếu X, thì Y", trong đó X và Y là các tập hợp (tập các mục) và được gọi lần lượt là phần đầu và phần cuối của quy luật. Mỗi quy luật còn có một giá trị đánh giá, bao gồm hỗ trợ (support) và độ tin cậy (confidence). Hỗ trợ là tỷ lệ của số lần xuất hiện của cả X và Y trong tập dữ liệu, trong khi độ tin cậy là xác suất của Y xuất hiện khi X xuất hiện.

Quy luật liên kết có thể được áp dụng trong nhiều lĩnh vực khác nhau, ví dụ như quản lý bán hàng, phân tích hành vi khách hàng, phân tích ngành công nghiệp, và nhiều lĩnh vực khác. Chúng có thể giúp tìm ra các mẫu và quy tắc tiềm năng, nhằm cung cấp thông tin hữu ích về mối quan hệ giữa các mục trong tập dữ liệu, từ đó giúp đưa ra các quyết định và chiến lược kinh doanh.

Trong đồ án này em sẽ sử dụng 2 thuật toán là FpGrowth và Apriori để tìm ra các rules của dataset, để tìm ra **xu hướng mua hàng theo độ tuổi, mặt hàng và trung tâm mua sắm**

- **Apriori**

Apriori là một thuật toán phổ biến và đơn giản để khai thác quy luật liên kết. Thuật toán này dựa trên nguyên tắc apriori, nghĩa là nếu một tập hợp con của một tập hợp lớn là phổ biến, thì tập hợp lớn đó cũng phải phổ biến. Apriori sử dụng một kỹ thuật gọi là tìm kiếm theo chiều rộng (breadth-first search) để tạo và kiểm tra các mẫu phổ biến. Quá trình khai thác quy luật bằng Apriori bao gồm ba bước chính: tạo các tập

phổ biến ban đầu, kết hợp các tập phổ biến để tạo các tập hợp lớn hơn, và kiểm tra tính phổ biến của các tập hợp mới.

- **FPGrowth**

FPGrowth (Frequent Pattern Growth) là một thuật toán hiệu quả để khai thác quy luật liên kết. FPGrowth dựa trên cấu trúc dữ liệu cây FP-Tree (Frequent Pattern Tree). Đầu tiên, FPGrowth xây dựng FP-Tree bằng cách quét dữ liệu một lần và xây dựng một cây tần suất dựa trên các mẫu phổ biến. Sau đó, FPGrowth sử dụng cây FP-Tree để tìm kiếm các quy luật liên kết. Quá trình tạo cây FP-Tree và khai thác quy luật được thực hiện trong cùng một lần quét dữ liệu, giúp giảm đáng kể thời gian xử lý so với thuật toán Apriori. FPGrowth cũng sử dụng đệ quy để khai thác các quy luật trong các cây con.

Tóm lại, cả FPGrowth và Apriori đều là các thuật toán quan trọng và phổ biến trong khai thác quy luật liên kết. FPGrowth hiệu quả hơn Apriori trong việc xử lý dữ liệu lớn và giảm thời gian tính toán nhờ cấu trúc dữ liệu FP-Tree. Tuy nhiên, Apriori vẫn có ý nghĩa và được sử dụng trong nhiều ứng dụng khai thác dữ liệu.

2.1. Tiền xử lý dữ liệu

Do dataset thuộc tính ‘age’ sẽ chỉ rõ số tuổi (18, 20 hay 22,...) điều này làm cho các rules được đưa ra có độ tin cậy không cao, hoặc thậm chí là không có. Do đó em đã chia nhóm độ tuổi ra thành một thuộc tính riêng có tên là ‘AgeGroup’ để phân vùng 1 nhóm tuổi tốt hơn đồng thời sẽ khiến cho các rules có độ tin cậy cao hơn.

```
Shopping=transactions.assign(AgeGroup=None)

Shopping.loc[(Shopping['age'] > 0) & (Shopping['age'] < 10), 'AgeGroup'] = '1-10'
Shopping.loc[(Shopping['age'] >= 10) & (Shopping['age'] < 20), 'AgeGroup'] = '10-20'
Shopping.loc[(Shopping['age'] >= 20) & (Shopping['age'] < 30), 'AgeGroup'] = '20-30'
Shopping.loc[(Shopping['age'] >= 30) & (Shopping['age'] < 40), 'AgeGroup'] = '30-40'
Shopping.loc[(Shopping['age'] >= 40) & (Shopping['age'] < 50), 'AgeGroup'] = '40-50'
Shopping.loc[(Shopping['age'] >= 50) & (Shopping['age'] < 60), 'AgeGroup'] = '50-60'
Shopping.loc[(Shopping['age'] >= 60) & (Shopping['age'] < 70), 'AgeGroup'] = '60-70'
Shopping.loc[Shopping['age'] >= 70, 'AgeGroup'] = 'Elderly'
Shopping
```

✓ 0.0s

	invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall	AgeGroup
0	I138884	C241288	Female	28	Clothing	5	1500.40	Credit Card	5/8/2022	Kanyon	20-30
1	I317333	C111565	Male	21	Shoes	3	1800.51	Debit Card	12/12/2021	Forum Istanbul	20-30
2	I127801	C266599	Male	20	Clothing	1	300.08	Cash	9/11/2021	Metrocity	20-30
3	I173702	C988172	Female	66	Shoes	5	3000.85	Credit Card	16/05/2021	Metropol AVM	60-70
4	I337046	C189076	Female	53	Books	4	60.60	Cash	24/10/2021	Kanyon	50-60

Nhóm em đã chia độ tuổi theo mỗi 10 tuổi. Như kết quả bên trên thì khách hàng có độ tuổi 28 sẽ có nhóm tuổi 20-30, khách hàng 66 tuổi sẽ có nhóm tuổi 60-70

Do để thực hiện được 2 thuật toán FpGrowth và Apriori thì tập dataset hoặc dataframe phải ở dạng nhị phân hoặc bool. Do đó đầu tiên để tìm các rules cho ‘AgeGroup ‘ - ‘Category’ và ‘Category’- ‘Shopping mall’ của thì em sẽ lấy 2 nhóm thuộc tính trên thành 2 dataframe mới và sau đó sẽ sử dụng one hot encoding để tách mỗi giá trị trong thuộc tính thành 1 attribute mới và chuyển chúng về dạng **bool**

- ‘AgeGroup’ và ‘Category’

	AgeGroup_10-20	AgeGroup_20-30	AgeGroup_30-40	AgeGroup_40-50	AgeGroup_50-60	AgeGroup_60-70	category_Books	category_Clothing	category_Cosmetics	category_Food & Beverage	category_Shoes	category_Sou
0	False	True	False	False	False	False	False	True	False	False	False	
1	False	True	False	False	False	False	False	False	False	False	True	
2	False	True	False	False	False	False	False	True	False	False	False	
3	False	False	False	False	False	True	False	False	False	False	True	
4	False	False	False	False	True	False	True	False	False	False	False	
5	False	True	False	False	False	False	False	True	False	False	False	
6	False	False	False	True	False	False	False	False	True	False	False	
7	False	False	True	False	False	False	False	True	False	False	False	
8	False	False	False	False	False	True	False	True	False	False	False	
9	False	False	False	False	False	True	False	True	False	False	False	

- ‘Category’ và ‘Shopping mall’

	shopping_mall_Cevahir AVM	shopping_mall_Emaar Square Mall	shopping_mall_Forum Istanbul	shopping_mall_Istinye Park	shopping_mall_Kanyon	shopping_mall_Mall of Istanbul	shopping_mall_Metrocity	shopping_mall_Metropol AVM	shopping_mall_Metropol AVM
0	False	False	False	False	True	False	False	False	False
1	False	False	True	False	False	False	False	False	False
2	False	False	False	False	False	False	True	False	False
3	False	False	False	False	False	False	False	False	True
4	False	False	False	False	True	False	False	False	False
5	False	False	True	False	False	False	False	False	False
6	False	False	False	True	False	False	False	False	False
7	False	False	False	False	False	True	False	False	False
8	False	False	False	False	False	False	True	False	False
9	False	False	False	False	True	False	False	False	False

Ví dụ nếu như khách hàng mua giày dép ở trung tâm **Kanyon**. Thì cột ‘shopping_mall_Kanyon’ và ‘category_shoes’ sẽ là true ở transactions của khách hàng đó, còn lại sẽ là false

2.2 Sử dụng thuật toán FpGrowth

Tiếp theo đó em sẽ build 1 model dựa trên thuật toán FpGrowth để tìm ra các itemset có mức support hơn mức chỉ định.

Mức Min Support em đặt ra là **0.02 (2%)**. Điều này là do sau khi thực hiện tính toán ngẫu nhiên tần suất xuất hiện của một vài itemset. Em nhận ra là mức con số kết quả tính được không quá **0.07 (7%)**. Do đó nhóm em đã đặt ra mức Support tối thiểu như trên để vừa có thể đảm vào các rules có độ tin cậy cao nhất xuất hiện, đồng thời có thể tìm ra các rules tiềm năng khác

- ‘AgeGroup’ và ‘Category’

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
6	(AgeGroup_40-50)	(category_Clothing)	0.192576	0.346753	0.067165	0.348770	1.005818	0.000389	1.003098	0.007164
0	(AgeGroup_20-30)	(category_Clothing)	0.193682	0.346753	0.067466	0.348336	1.004566	0.000307	1.002430	0.005637
2	(AgeGroup_60-70)	(category_Clothing)	0.191470	0.346753	0.066391	0.346742	0.999967	-0.000002	0.999983	-0.000040
4	(AgeGroup_50-60)	(category_Clothing)	0.190344	0.346753	0.065667	0.344990	0.994915	-0.000336	0.997308	-0.006273
18	(AgeGroup_30-40)	(category_Clothing)	0.193923	0.346753	0.066662	0.343755	0.991354	-0.000581	0.995432	-0.010704
30	(category_Toys)	(AgeGroup_20-30)	0.101421	0.193682	0.020451	0.201646	1.041119	0.000808	1.009976	0.043953
21	(category_Food & Beverage)	(AgeGroup_30-40)	0.148567	0.193923	0.029480	0.198430	1.023241	0.000670	1.005623	0.026676
16	(category_Cosmetics)	(AgeGroup_30-40)	0.151794	0.193923	0.029691	0.195602	1.008657	0.000255	1.002087	0.010119
14	(category_Cosmetics)	(AgeGroup_50-60)	0.151794	0.190344	0.029561	0.194741	1.023101	0.000667	1.005460	0.026620

Ta có thể thấy được với mặt hàng ‘**clothing**’ được chọn mua ở mọi lứa tuổi với mức độ tin tưởng **confidence** không quá chênh lệch nhau (~0.34)

- ‘**Category**’ và ‘**Shopping mall**’

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
6	(shopping_mall_Metrocity)	(category_Clothing)	0.150930	0.346753	0.052968	0.350943	1.012083	0.000632	1.006455	0.014061
18	(shopping_mall_Mall of Istanbul)	(category_Clothing)	0.200519	0.346753	0.069608	0.347139	1.001115	0.000077	1.000592	0.001393
10	(shopping_mall_Metropol AVM)	(category_Clothing)	0.102165	0.346753	0.035442	0.346915	1.000467	0.000017	1.000248	0.000519
0	(shopping_mall_Kanyon)	(category_Clothing)	0.199312	0.346753	0.068773	0.345054	0.995100	-0.000339	0.997406	-0.006112
16	(shopping_mall_Istinye Park)	(category_Clothing)	0.098344	0.346753	0.033713	0.342807	0.988622	-0.000388	0.993997	-0.012603
13	(category_Cosmetics)	(shopping_mall_Mall of Istanbul)	0.151794	0.200519	0.030667	0.202027	1.007521	0.000229	1.001890	0.008801
4	(category-Shoes)	(shopping_mall_Mall of Istanbul)	0.100888	0.200519	0.020340	0.201615	1.005464	0.000111	1.001372	0.006044
3	(category-Shoes)	(shopping_mall_Kanyon)	0.100888	0.199312	0.020280	0.201017	1.008551	0.000172	1.002133	0.009430

Điều tương tự cũng xảy ra với mặt hàng này khi nó được chọn mua nhiều ở các trung tâm mua sắm với mức độ tin cậy gần như tương đương (~0.34)

Ngoài ra thì có một rules tiềm năng là mặt hàng ‘**Shoes**’ được mua nhiều ở mall **Kanyon và Istanbul** với mức độ tin cậy khoảng 0.2

Kết luận:

Từ kết quả có được như trên ta có thể thấy rằng, mặt hàng ‘**Clothing**’ được chọn mua ở mọi lứa tuổi.

Ngoài ra thì nhóm khách hàng mua mặt hàng ‘**Toy**’ thường ở độ tuổi 20-30 và mặt hàng ‘**Food & Beverage**’ cũng có xu hướng được mua ở lứa tuổi 30-40

Ta cũng có thể thấy được rằng mặt hàng ‘**Clothing**’ được mua nhiều nhất ở hầu hết các khu trung tâm mua sắm. Cho thấy đây là mặt hàng bán chạy được mua rộng rãi ở mọi lứa tuổi. Đi kèm với đó mặt hàng ‘**Shoes**’ cũng được mua nhiều ở trung tâm mua sắm **Kanyon và Istanbul**. Từ đó ta có thể thấy các trung tâm mua sắm này được tin tưởng để mua các mặt hàng may mặc

2.3 Sử dụng thuật toán Apriori

Sau khi đưa ra các kết luận trên. Để đảm bảo tính chính xác và độ tin cậy, em đã thực hiện tìm ra các rules bằng một thuật toán khác là Apriori để so sánh với kết quả vừa được đưa ra bởi thuật toán FpGrowth

Bởi vì mục đích để so sánh và vì nó cùng là 2 thuật toán của Association Rules. Nên em sẽ sử dụng lại model được dựng nên ở bước tiền xử lý dữ liệu của thuật toán FpGrowth để sử dụng cho thuật toán Apriori này

- ‘AgeGroup’ và ‘Category’

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
6	(AgeGroup_40-50)	(category_Clothing)	0.192576	0.346753	0.067165	0.348770	1.005818	0.000389	1.003098	0.007164
0	(AgeGroup_20-30)	(category_Clothing)	0.193682	0.346753	0.067466	0.348336	1.004566	0.000307	1.002430	0.005637
2	(AgeGroup_60-70)	(category_Clothing)	0.191470	0.346753	0.066391	0.346742	0.999967	-0.000002	0.999983	-0.000040
4	(AgeGroup_50-60)	(category_Clothing)	0.190344	0.346753	0.065667	0.344990	0.994915	-0.000336	0.997308	-0.006273
18	(AgeGroup_30-40)	(category_Clothing)	0.193923	0.346753	0.066662	0.343755	0.991354	-0.000581	0.995432	-0.010704
30	(category_Toys)	(AgeGroup_20-30)	0.101421	0.193682	0.020451	0.201646	1.041119	0.000808	1.009976	0.043953
21	(category_Food & Beverage)	(AgeGroup_30-40)	0.148567	0.193923	0.029480	0.198430	1.023241	0.000670	1.005623	0.026676
16	(category_Cosmetics)	(AgeGroup_30-40)	0.151794	0.193923	0.029691	0.195602	1.008657	0.000255	1.002087	0.010119
14	(category_Cosmetics)	(AgeGroup_50-60)	0.151794	0.190344	0.029561	0.194741	1.023101	0.000667	1.005460	0.026620

- ‘Category’ và ‘Shopping mall’

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
6	(shopping_mall_Metrocity)	(category_Clothing)	0.150930	0.346753	0.052968	0.350943	1.012083	0.000632	1.006455	0.014061
18	(shopping_mall_Mall of Istanbul)	(category_Clothing)	0.200519	0.346753	0.069608	0.347139	1.001115	0.000077	1.000592	0.001393
10	(shopping_mall_Metropol AVM)	(category_Clothing)	0.102165	0.346753	0.035442	0.346915	1.000467	0.000017	1.000248	0.000519
0	(shopping_mall_Kanyon)	(category_Clothing)	0.199312	0.346753	0.068773	0.345054	0.995100	-0.000339	0.997406	-0.006112
16	(shopping_mall_Istinye Park)	(category_Clothing)	0.098344	0.346753	0.033713	0.342807	0.988622	-0.000388	0.993997	-0.012603
13	(category_Cosmetics)	(shopping_mall_Mall of Istanbul)	0.151794	0.200519	0.030667	0.202027	1.007521	0.000229	1.001890	0.008801
4	(category-Shoes)	(shopping_mall_Mall of Istanbul)	0.100888	0.200519	0.020340	0.201615	1.005464	0.000111	1.001372	0.006044
3	(category-Shoes)	(shopping_mall_Kanyon)	0.100888	0.199312	0.020280	0.201017	1.008551	0.000172	1.002133	0.009430

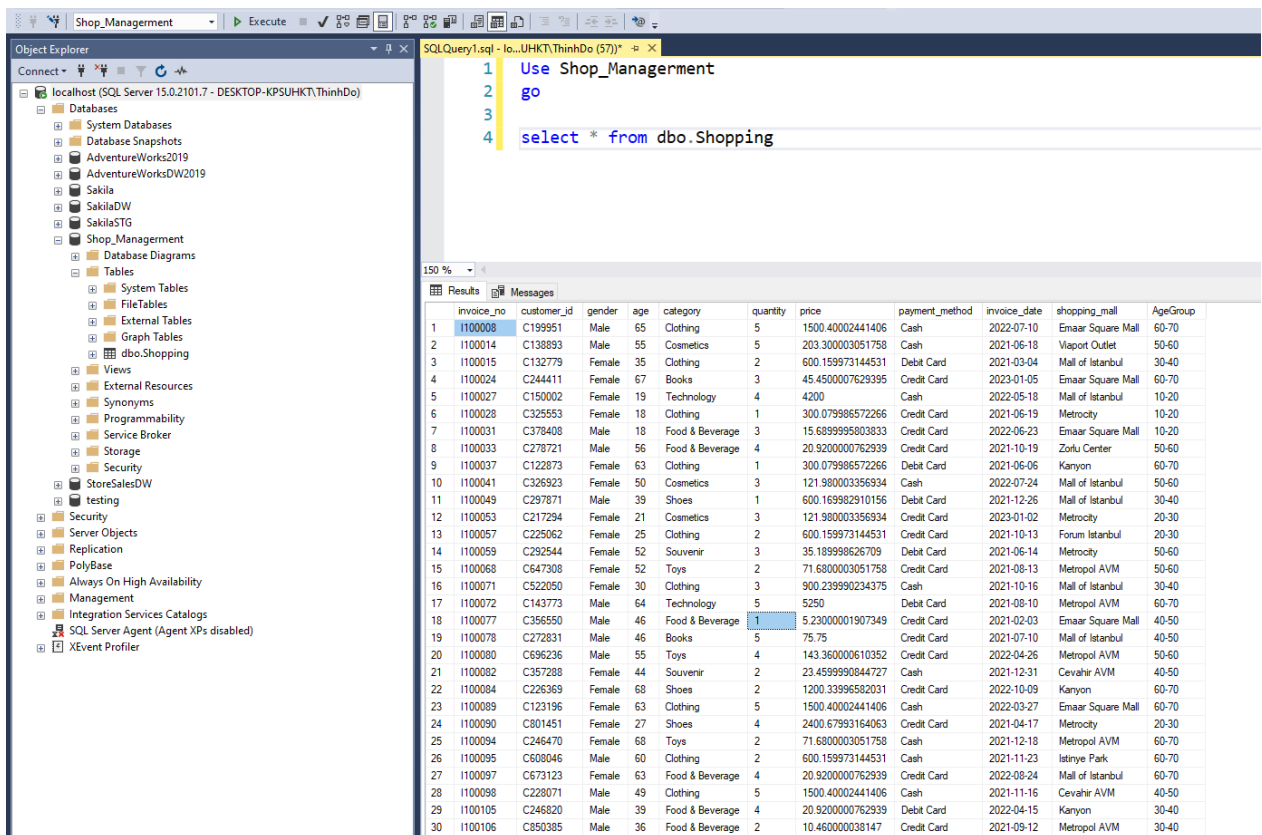
Và kết quả này cho thấy 2 kết quả được đưa ra từ 2 thuật toán Apriori và FpGrowth là tương đồng nhau. Nên có thể nói kết quả được đưa ra là có độ chính xác tốt và có thể tin cậy được

3. Association rules trên SSAS

3.1. Chuẩn bị dữ liệu

Để có thể thực hiện SSAS thì việc đưa dữ liệu từ file csv trực tiếp vào để phân tích là không thể.

Trước tiên ta cần phải nhập dữ liệu từ file CSV vào trong SQL Server để có được dữ liệu như bảng sau.



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the database structure for 'Shop_Management'. The main window shows a SQL query window with the following code:

```
1 Use Shop_Management
2 go
3
4 select * from dbo.Shopping
```

The query results are displayed in a table with the following columns: invoice_no, customer_id, gender, age, category, quantity, price, payment_method, invoice_date, shopping_mall, and AgeGroup. The table contains 30 rows of data.

invoice_no	customer_id	gender	age	category	quantity	price	payment_method	invoice_date	shopping_mall	AgeGroup	
1	I100008	C199951	Male	65	Clothing	5	1500.40002441406	Cash	2022-07-10	Emaar Square Mall	60-70
2	I100014	C138893	Male	55	Cosmetics	5	203.300003051758	Cash	2021-06-18	Viaport Outlet	50-60
3	I100015	C132779	Female	35	Clothing	2	600.159973144531	Debit Card	2021-03-04	Mall of Istanbul	30-40
4	I100024	C244411	Female	67	Books	3	45.4500007629395	Credit Card	2023-01-05	Emaar Square Mall	60-70
5	I100027	C150002	Female	19	Technology	4	4200	Cash	2022-05-18	Mall of Istanbul	10-20
6	I100028	C325553	Female	18	Clothing	1	300.079986572266	Credit Card	2021-06-19	Metrocity	10-20
7	I100031	C378408	Male	18	Food & Beverage	3	15.6899995803833	Credit Card	2022-06-23	Emaar Square Mall	10-20
8	I100033	C278721	Male	56	Food & Beverage	4	20.9200000762939	Credit Card	2021-10-19	Zorlu Center	50-60
9	I100037	C122873	Female	63	Clothing	1	300.079986572266	Debit Card	2021-06-06	Kanyon	60-70
10	I100041	C326923	Female	50	Cosmetics	3	121.980003356934	Cash	2022-07-24	Mall of Istanbul	50-60
11	I100049	C297871	Male	39	Shoes	1	600.169982910156	Debit Card	2021-12-26	Mall of Istanbul	30-40
12	I100053	C217294	Female	21	Cosmetics	3	121.980003356934	Credit Card	2023-01-02	Metrocity	20-30
13	I100057	C225062	Female	25	Clothing	2	600.159973144531	Credit Card	2021-10-13	Forum Istanbul	20-30
14	I100059	C292544	Female	52	Souvenir	3	35.189998626709	Debit Card	2021-06-14	Metrocity	50-60
15	I100068	C647308	Female	52	Toys	2	71.6800003051758	Credit Card	2021-08-13	Metrocity AVM	50-60
16	I100071	C522050	Female	30	Clothing	3	900.239990234375	Cash	2021-10-16	Mall of Istanbul	30-40
17	I100072	C143773	Male	64	Technology	5	5250	Debit Card	2021-08-10	Metrocity AVM	60-70
18	I100077	C356550	Male	46	Food & Beverage	1	5.23000001907349	Credit Card	2021-02-03	Emaar Square Mall	40-50
19	I100078	C272831	Male	46	Books	5	75.75	Credit Card	2021-07-10	Mall of Istanbul	40-50
20	I100080	C696236	Male	55	Toys	4	143.360000610352	Credit Card	2022-04-26	Metrocity AVM	50-60
21	I100082	C357288	Female	44	Souvenir	2	23.459999844727	Cash	2021-12-31	Cevahir AVM	40-50
22	I100084	C226369	Female	68	Shoes	2	1200.33996582031	Credit Card	2022-10-09	Kanyon	60-70
23	I100089	C123196	Female	63	Clothing	5	1500.40002441406	Cash	2022-03-27	Emaar Square Mall	60-70
24	I100090	C801451	Female	27	Shoes	4	2400.67993164063	Credit Card	2021-04-17	Metrocity	20-30
25	I100094	C246470	Female	68	Toys	2	71.6800003051758	Cash	2021-12-18	Metrocity AVM	60-70
26	I100095	C608046	Male	60	Clothing	2	600.159973144531	Cash	2021-11-23	Itinye Park	60-70
27	I100097	C673123	Female	63	Food & Beverage	4	20.9200000762939	Credit Card	2022-08-24	Mall of Istanbul	60-70
28	I100098	C228071	Male	49	Clothing	5	1500.40002441406	Cash	2021-11-16	Cevahir AVM	40-50
29	I101105	C246820	Male	39	Food & Beverage	4	20.9200000762939	Debit Card	2022-04-15	Kanyon	30-40
30	I101106	C850385	Male	36	Food & Beverage	2	10.460000038147	Credit Card	2021-09-12	Metrocity AVM	30-40

Sau khi đã có dữ liệu trong database SQL Server ta tiến hành tạo project Analysis Services Multidimensional and Data Mining Project

Configure your new project

Analysis Services Multidimensional and Data Mining Project

Project name
DataMining

Location
C:\Users\ThinkDo\source\repos

Solution name ⓘ
DataMining

☐ Place solution and project in the same directory

Back Create

Tiếp đến ta thêm Data Sources từ Shop_Management từ SQL Server

Data Source Wizard

Select how to define the connection
You can select from a number of ways in which your data source will define its connection string.

☐ Create a data source based on another object

☒ Create a data source based on an existing or new connection

Data connections:

localhost.Shop_Management

Data connection properties:

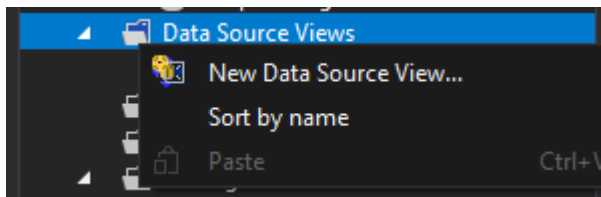
Property	Value
Data Source	localhost
Initial Catalog	Shop_Management
Integrated Se...	SSPI
Provider	MSOLEDBSQL.1

New... Delete

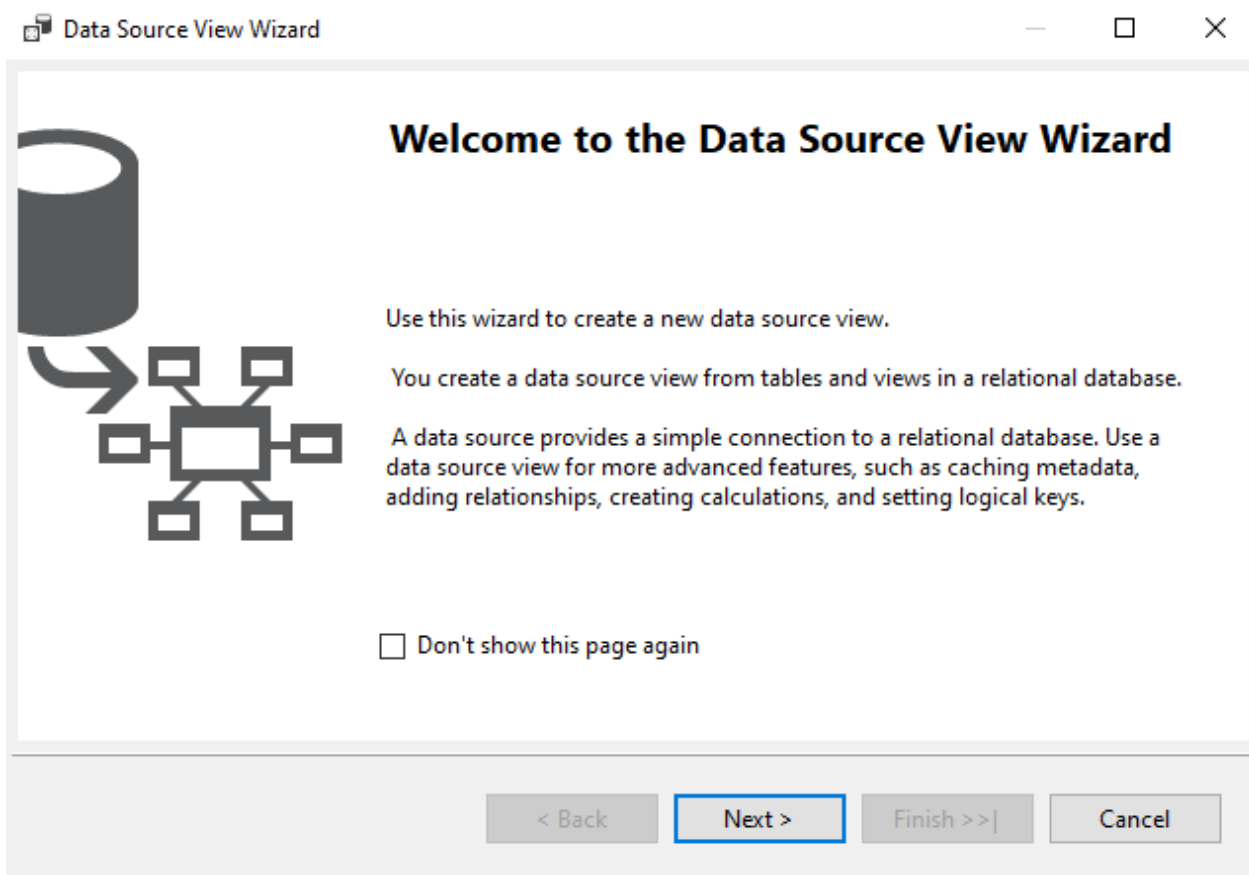
< Back Next > Finish >>| Cancel

Sau khi hoàn tất việc kết nối SSAS đến Database ta tiến hành việc tạo Data Source View để chọn ra bảng dữ liệu mà ta đã thêm vào Database lúc trước như sau

Ở **Data Source Views** ta chọn **New Data Source View...**

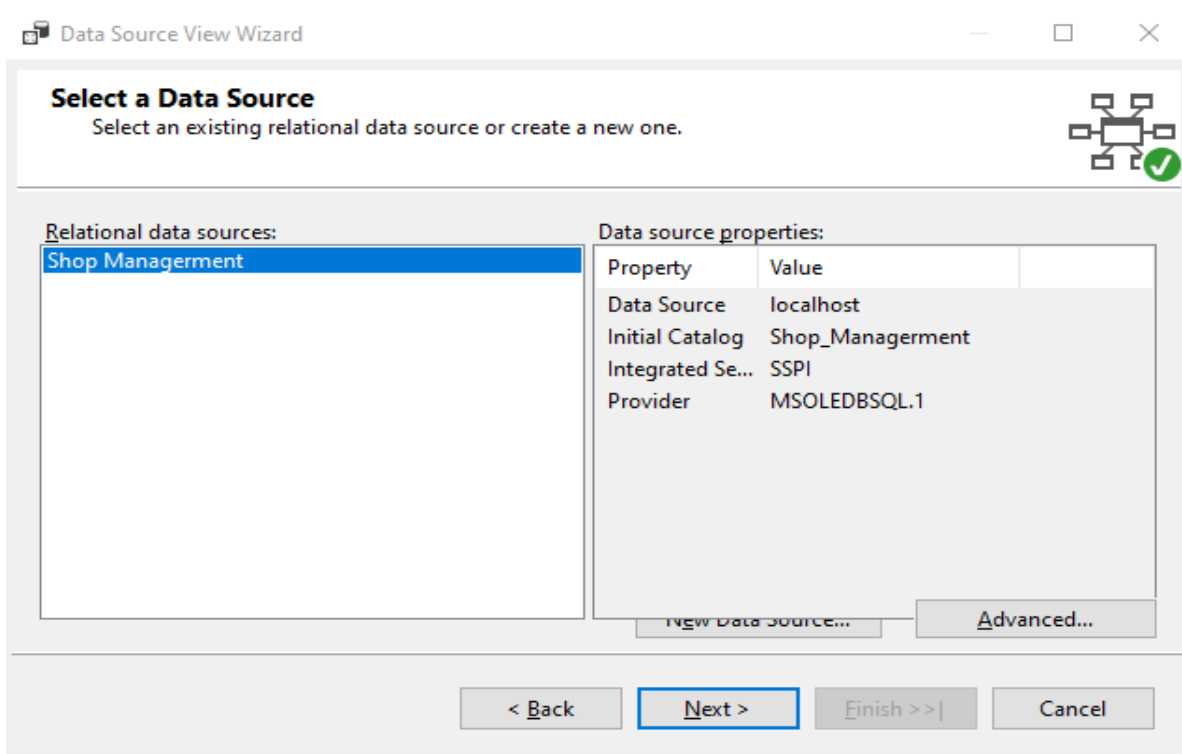


Tiếp đến ta chọn next

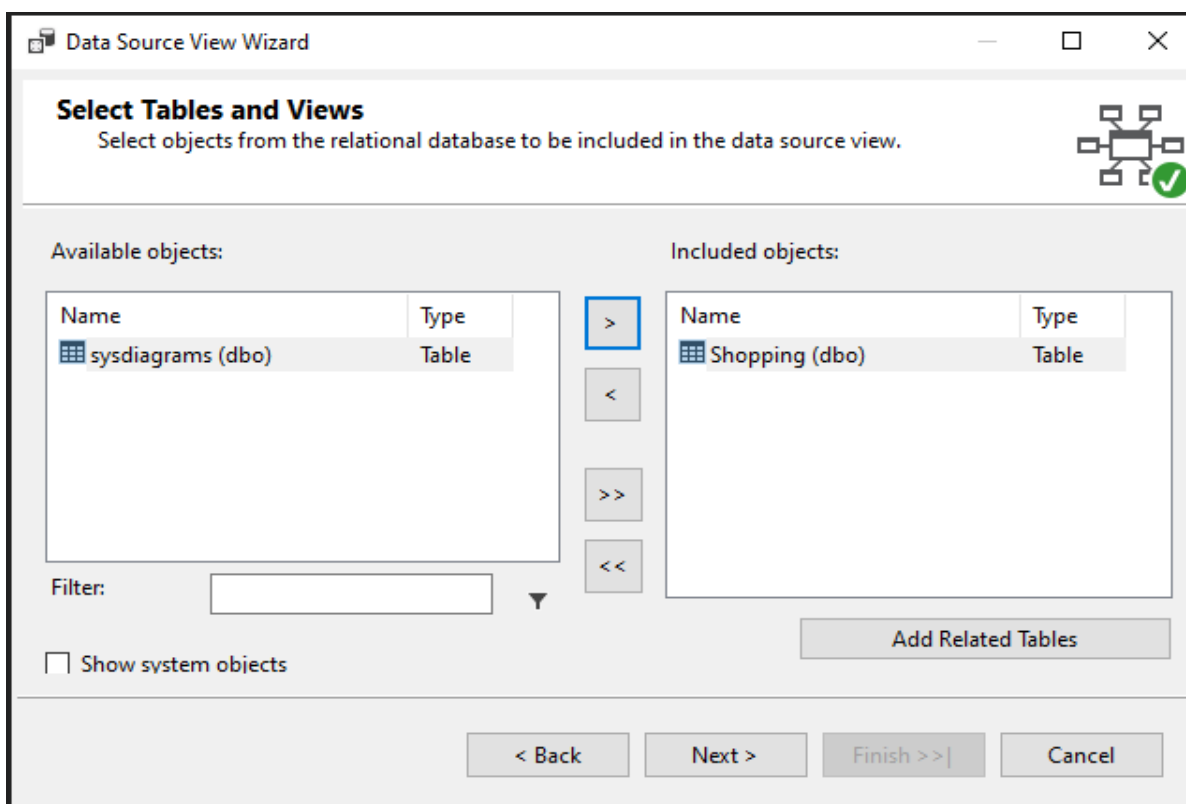


Ở đây ta sẽ thấy các connection đến các Database hiện có

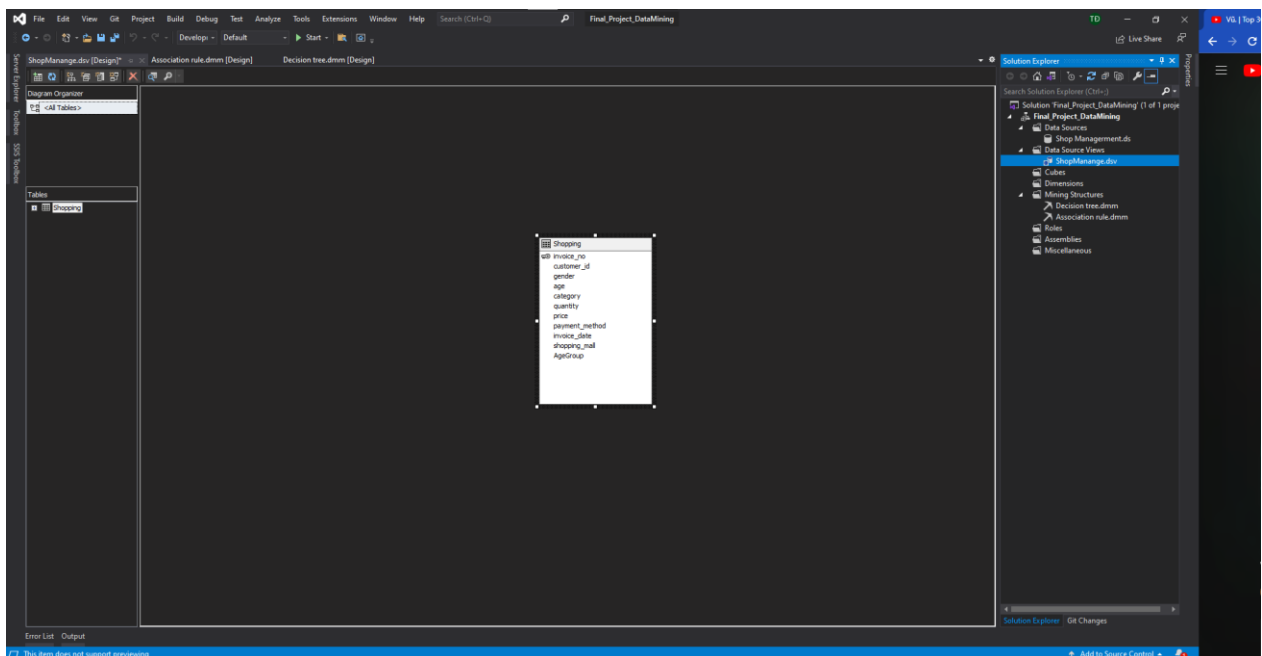
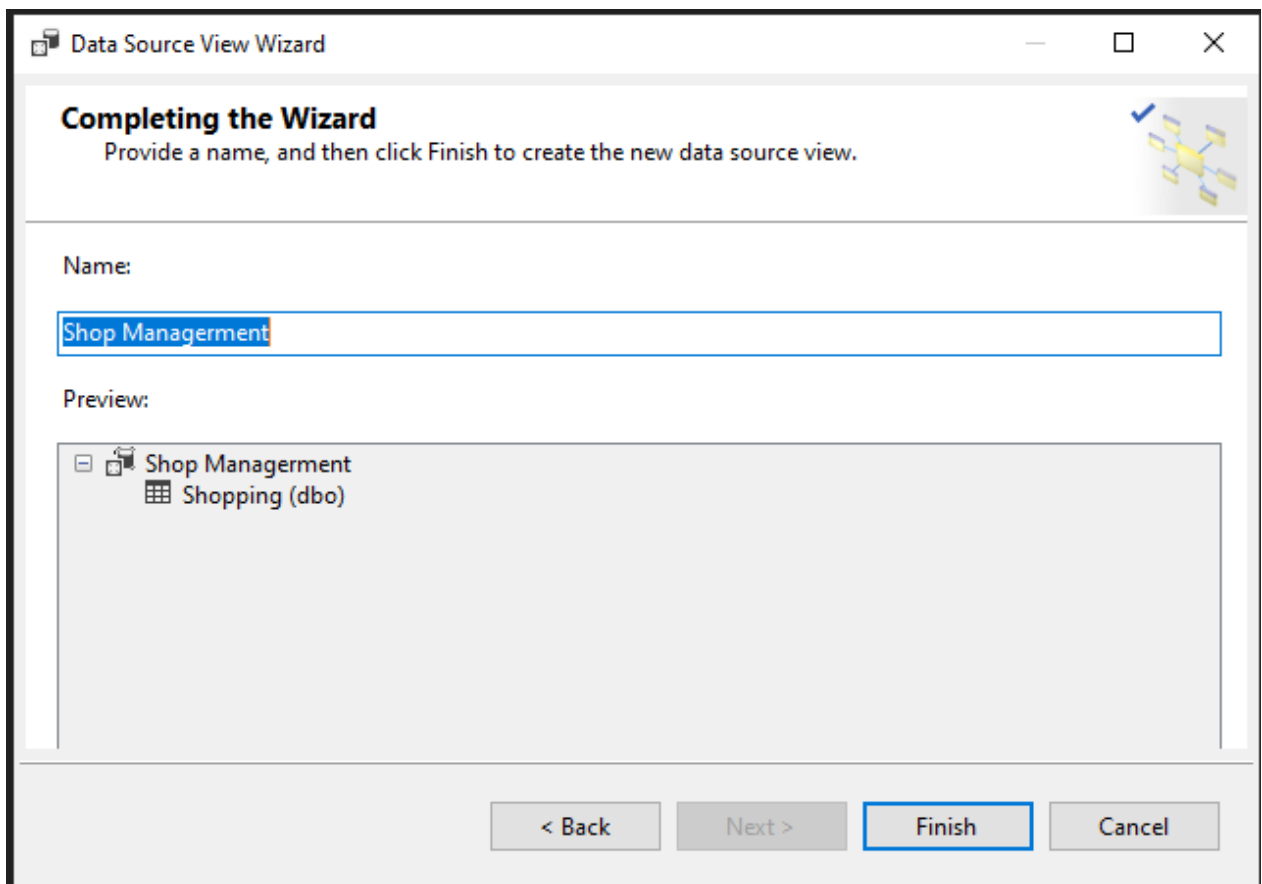
Ta chọn **Shop_Management** rồi nhấn **Next**



Ở đây ta chọn bảng **Shopping** rồi nhấn “>” sau đó ta nhấn **next**



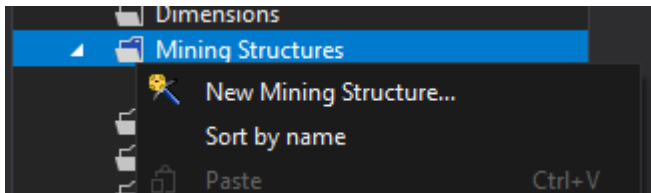
Tới đây ta đặt tên cho View là **ShopManage** rồi chọn **Finish**



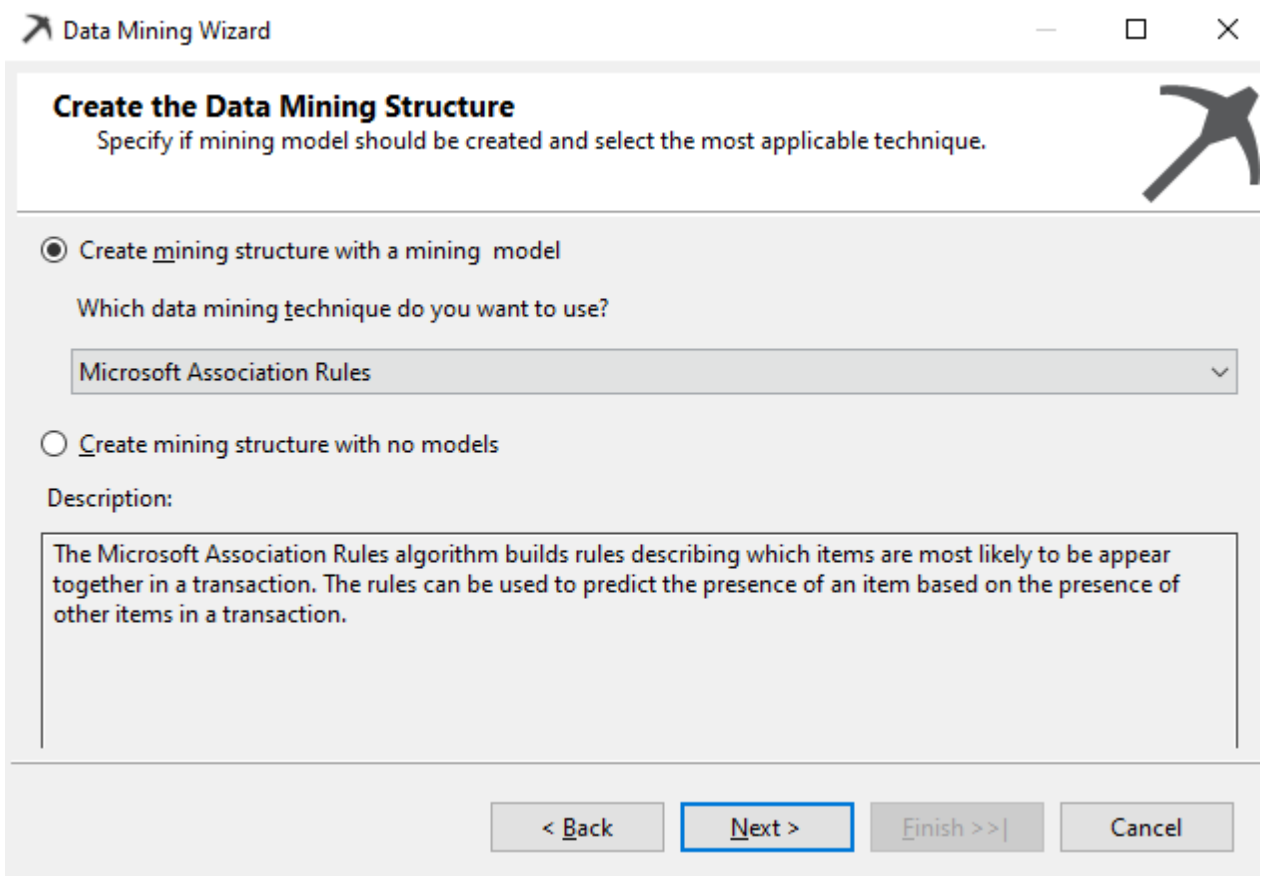
3.2. Mining Structure on SSAS

- **Tạo Mining Structure**

Đầu tiên ta click chuột phải vào Folder Mining Structures rồi chọn New Mining Structure



Nhấn next liên tục cho đến khi bảng sau hiện ra rồi ta chọn thuật toán là **Microsoft Association rules**



Tiếp tục nhấn next cho đến khi bảng sau hiện ra và chọn như bảng bên

Data Mining Wizard

Specify the Training Data
Specify the columns used in your analysis.

Mining model structure:

<input type="checkbox"/>	Tables/Columns	Key	<input type="checkbox"/> Input	<input type="checkbox"/> Predict...
-	Shopping			
<input type="checkbox"/>	age	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	AgeGroup	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	category	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	customer_id	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	gender	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	invoice_date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	invoice_no	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	payment_method	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	price	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	quantity	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	shopping_mall	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Recommend inputs for currently selected predictable:

Tiếp tục nhấn next cho đến khi hiện ra bảng sau, ta tiến hành đặt tên cho bảng rồi chọn **Finish**

Data Mining Wizard

Completing the Wizard
Completing the Data Mining Wizard by providing a name for the mining structure.

Mining structure name:

Mining model name:

☐ Allow drill through

Preview:

- Shopping
 - Columns
 - Age Group
 - Category
 - Gender
 - Invoice No
 - Payment Method
 - Price
 - Quantity
 - Shopping Mall

< Back Next > **Finish** Cancel

Sau khi hoàn thành các bước trên ta được như sau

ShopManage.dsv [Design] Association rule.dmm [Design]

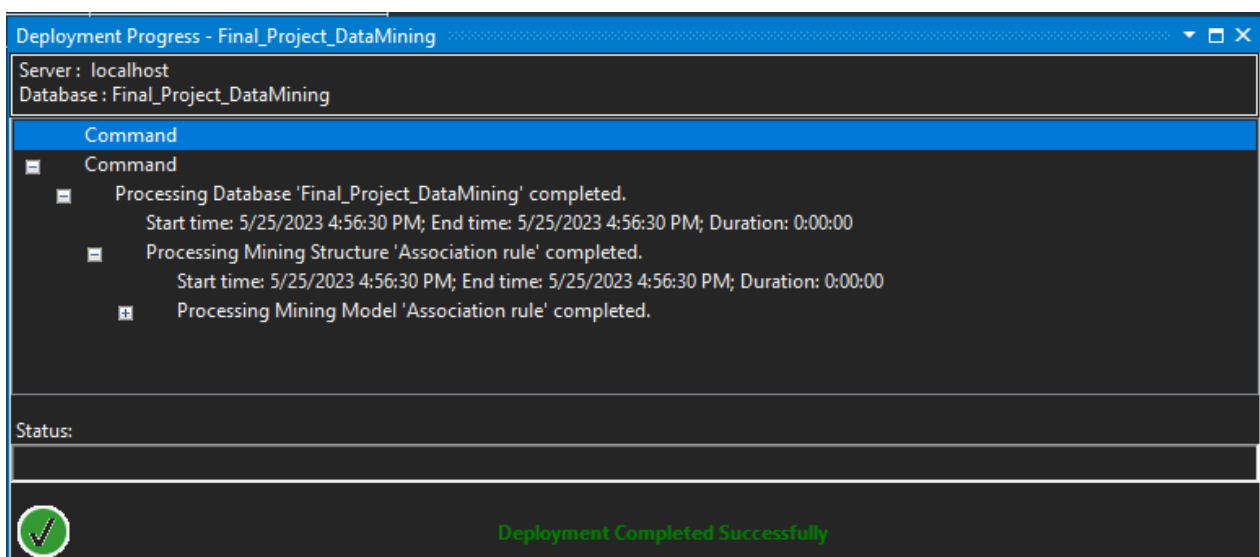
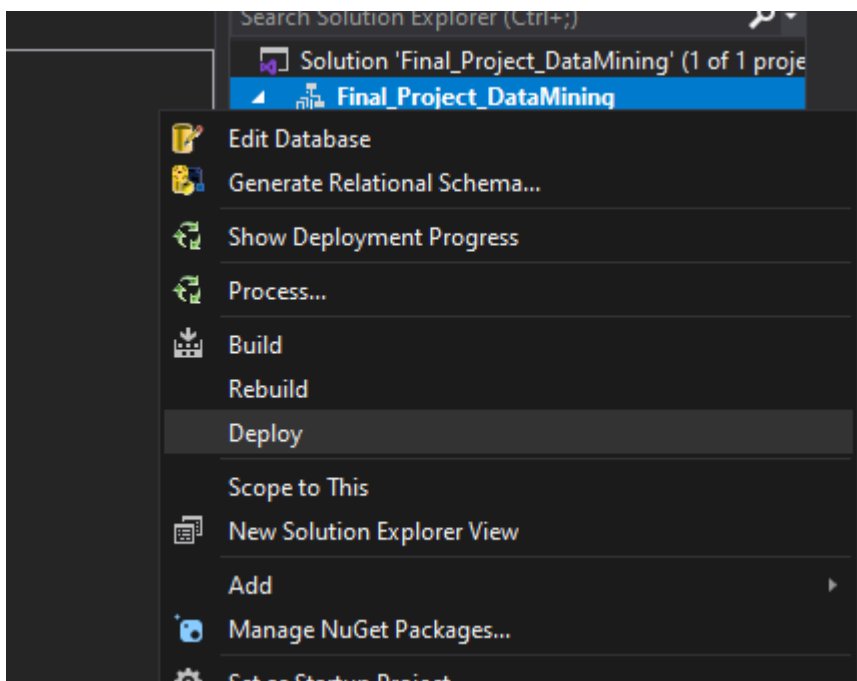
Mining Structure **Mining Models** Mining Model Viewer Mining Accuracy Chart Mining Model Prediction

Structure	Association rule
	Microsoft_Association_Rules
Age Group	Predict
Category	Predict
Gender	Predict
Invoice No	Key
Price	Input
Quantity	Predict
Shopping Mall	Predict

Solution Explorer

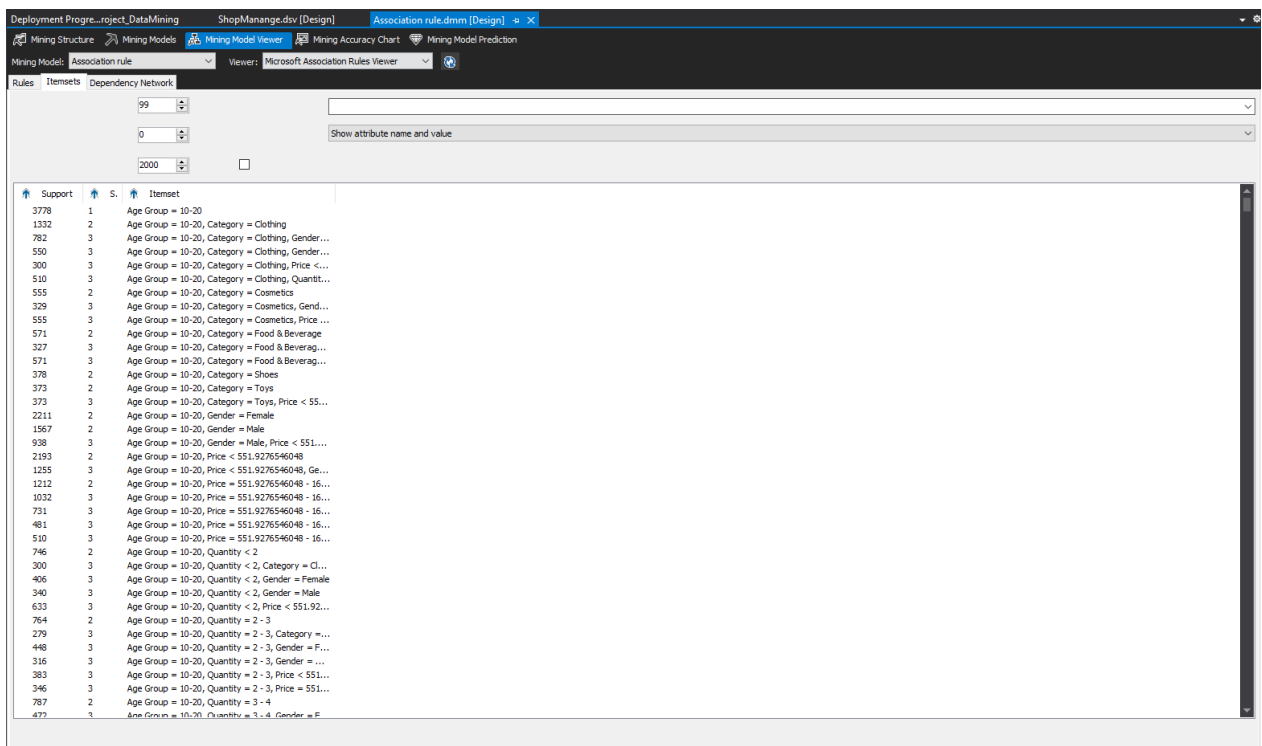
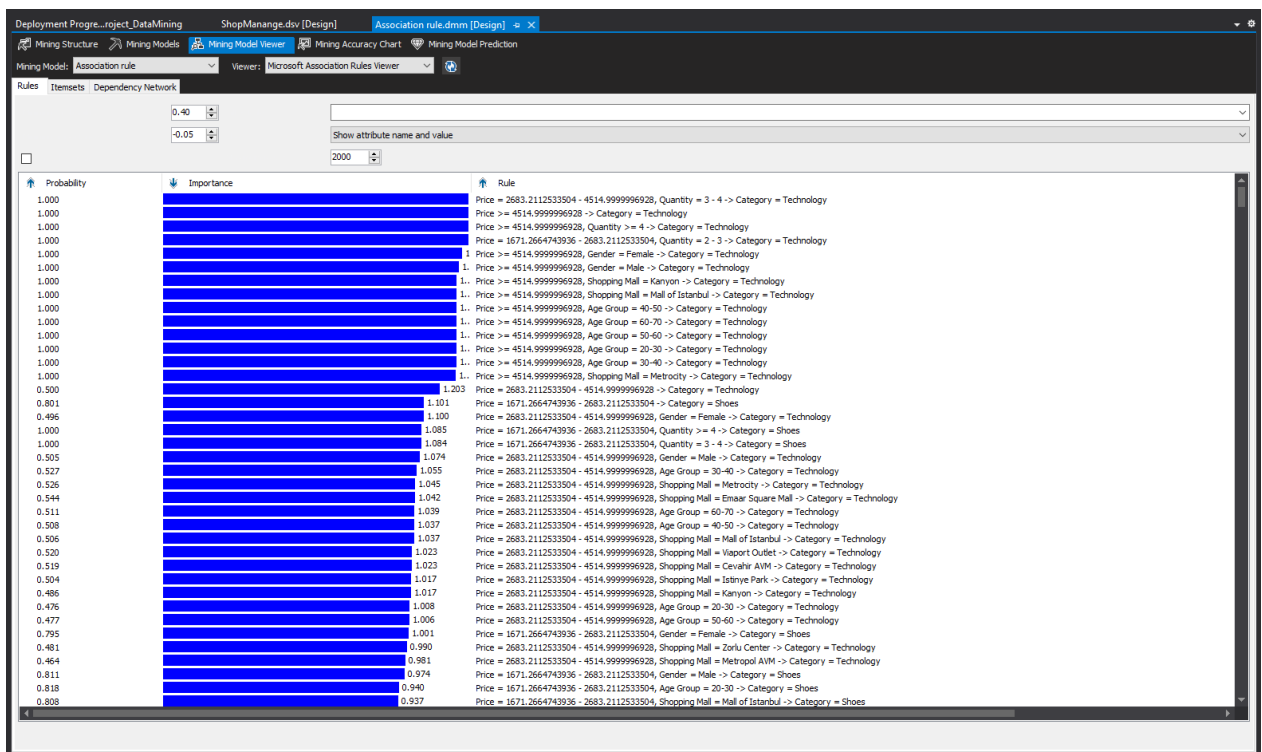
- Solution_Final_Project_DataMining (1 of 1 projects)
 - Final_Project_DataMining
 - Data Sources
 - Shop Management.dsv
 - Data Source Views
 - ShopManage.dsv
 - Cubes
 - Dimensions
 - Mining Structures**
 - Association rule.dmm
 - Roles
 - Assemblies
 - Miscellaneous

Nhấn chuột phải vào Project rồi chọn deploy để thực hiện khai phá dữ liệu



- **Khai phá dữ liệu**

Sau khi hoàn thành các bước trên ta có được bảng dữ liệu như sau



Sau khi tiến hành phân tích về dữ liệu nhóm chúng em có đặt ra một số nghiệp vụ và tiến hành trả lời các câu hỏi đó bằng thông tin mà bảng dữ liệu đem lại như sau.

Ở các nhóm độ tuổi khác nhau thì họ sẽ quan tâm đến những mặt hàng nào, họ thường lui đến những cửa hàng nào, số tiền mà họ chi ra cho việc mua sắm nằm ở khoảng nào.

Từ câu hỏi trên nhóm em bắt đầu phân tích 2 nhóm tuổi chính đó là 10-20 và 30-40 để có thể đưa ra kết luận nhanh chóng

Đầu tiên là ở nhóm tuổi 10-20:

1332	Age Group = 10-20, Category = Clothing
571	Age Group = 10-20, Category = Food & Beverage
555	Age Group = 10-20, Category = Cosmetics
378	Age Group = 10-20, Category = Shoes
373	Age Group = 10-20, Category = Toys
200	Age Group = 10-20, Category = Books
194	Age Group = 10-20, Category = Souvenir
175	Age Group = 10-20, Category = Technology

Ở nhóm độ tuổi này ta có thể thấy những nhóm mặt hàng mà độ tuổi này quan tâm theo mức độ giảm dần là quần áo, ăn uống, làm đẹp, giày, đồ chơi, sách và quà tặng

2193	Age Group = 10-20, Price < 551.9276546048
1212	Age Group = 10-20, Price = 551.9276546048 - 1671.2664743936

203	Age Group = 10-20, Price = 1671.2664743936 - 2683.2112533504
146	Age Group = 10-20, Price = 2683.2112533504 - 4514.9999996928

Về khoản thanh toán thì ta có thể thấy hơn 1 nửa trong tổng số các thanh toán đều có giá trị thấp hơn 551.9 nói chung đều là mức chi thấp

740	Age Group = 10-20, Shopping Mall = Mall of Istanbul
738	Age Group = 10-20, Shopping Mall = Kanyon
578	Age Group = 10-20, Shopping Mall = Metrocity
393	Age Group = 10-20, Shopping Mall = Istinye Park
374	Age Group = 10-20, Shopping Mall = Metropol AVM
205	Age Group = 10-20, Shopping Mall = Forum Istanbul
197	Age Group = 10-20, Shopping Mall = Viaport Outlet
193	Age Group = 10-20, Shopping Mall = Zorlu Center
190	Age Group = 10-20, Shopping Mall = Cevahir AVM
170	Age Group = 10-20, Shopping Mall = Emaar Square Mall

Ở nhóm độ tuổi này ta thấy họ thường có xu hướng đi 3 trung tâm mua sắm là Mall of Istanbul, Kanyon và Metrocity.

Tiếp đến là nhóm độ tuổi 30-40

6625	Age Group = 30-40, Category = Clothing
2948	Category = Cosmetics, Age Group = 30-40
2929	Category = Food & Beverage, Age Group = 30-40
1921	Category = Toys, Age Group = 30-40
1901	Category = Shoes, Age Group = 30-40
1019	Category = Technology, Age Group = 30-40
986	Category = Souvenir, Age Group = 30-40
939	Category = Books, Age Group = 30-40

Ở nhóm độ tuổi này ta bắt đầu thấy có sự khác biệt về mức độ mua hàng và mặt hàng họ mua họ bắt đầu quan tâm về mặt hàng công nghệ hơn và các mặt hàng họ mua cũng nhiều hơn so với độ tuổi 10-20

11085	Age Group = 30-40, Price < 551.9276546048
6198	Age Group = 30-40, Price = 551.9276546048 - 1671.2664743936
797	Price = 2683.2112533504 - 4514.9999996928, Age Group = 30-40
189	Price >= 4514.9999996928, Age Group = 30-40

Về hành vi thanh toán của người dùng đã bắt đầu có xu hướng chi tiêu nhiều hơn mặc dù so với tổng số thì không nhiều nhưng đây là phân khúc tốt để nhà bán hàng tiếp cận để bán các mặt hàng đắt tiền.

3849	Age Group = 30-40, Shopping Mall = Mall of Istanbul
3820	Age Group = 30-40, Shopping Mall = Kanyon
2851	Shopping Mall = Metrocity, Age Group = 30-40
1993	Shopping Mall = Metropol AVM, Age Group = 30-40
1880	Shopping Mall = Istinye Park, Age Group = 30-40
999	Shopping Mall = Zorlu Center, Age Group = 30-40
985	Shopping Mall = Cevahir AVM, Age Group = 30-40
976	Shopping Mall = Emaar Square Mall, Age Group = 30-40
960	Shopping Mall = Viaport Outlet, Age Group = 30-40
955	Shopping Mall = Forum Istanbul, Age Group = 30-40

Về độ phổ biến của các Mall top đầu cũng gần như tương tự như độ tuổi 10-20 nhưng ở các Mall khác lúc này đã có sự thay đổi về lượng người ra vào ở các Mall dưới top có lẽ là do nhu cầu mua sắm thay đổi nên thứ tự của chúng cũng có sự thay đổi.

Sau khi xem qua về 2 nhóm tuổi thì nhóm chúng em có đặt ra thêm câu hỏi giả định rằng, vậy lúc này mình muốn tăng doanh thu cho mặt hàng công nghệ thì liệu tệp khách hàng nào mình có thể tiếp cận, các Mall nào bán được nhiều nhất và bán được cho ai để có thể đưa ra được chiến lược kinh doanh hợp lý.

- **Shopping Mall**

1017	Shopping Mall = Mall of Istanbul
996	Shopping Mall = Kanyon
772	Shopping Mall = Metrocity
487	Shopping Mall = Istinye Park
464	Shopping Mall = Metropol AVM
262	Shopping Mall = Emaar Square Mall
256	Shopping Mall = Cevahir AVM
256	Shopping Mall = Viaport Outlet
250	Shopping Mall = Zorlu Center
234	Shopping Mall = Forum Istanbul

- **Age - Group**

1019	Age Group = 30-40
980	Age Group = 40-50
961	Age Group = 60-70
952	Age Group = 20-30
907	Age Group = 50-60
175	Age Group = 10-20

- **Price - Age**

420	Price = 2683.2112533504 - 4514.9999996928, Age Group = 30-40
400	Price = 2683.2112533504 - 4514.9999996928, Age Group = 40-50
382	Price = 2683.2112533504 - 4514.9999996928, Age Group = 20-30
380	Price = 2683.2112533504 - 4514.9999996928, Age Group = 60-70
355	Price = 2683.2112533504 - 4514.9999996928, Age Group = 50-60
219	Price = 1671.2664743936 - 2683.2112533504, Age Group = 30-40
205	Age Group = 20-30, Price = 551.9276546048 - 1671.2664743936
203	Price >= 4514.9999996928, Age Group = 40-50
201	Price >= 4514.9999996928, Age Group = 60-70
198	Price = 1671.2664743936 - 2683.2112533504, Age Group = 40-50
197	Price = 1671.2664743936 - 2683.2112533504, Age Group = 60-70
193	Price >= 4514.9999996928, Age Group = 50-60
191	Age Group = 30-40, Price = 551.9276546048 - 1671.2664743936
191	Price >= 4514.9999996928, Age Group = 20-30
189	Price >= 4514.9999996928, Age Group = 30-40
183	Age Group = 50-60, Price = 551.9276546048 - 1671.2664743936
183	Age Group = 60-70, Price = 551.9276546048 - 1671.2664743936

179	Age Group = 40-50, Price = 551.9276546048 - 1671.2664743936
176	Price = 1671.2664743936 - 2683.2112533504, Age Group = 50-60
174	Price = 1671.2664743936 - 2683.2112533504, Age Group = 20-30

- **Price - Shop**

411	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Mall of Istanbul
384	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Kanyon
312	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Metrocity
222	Price \geq 4514.9999996928, Shopping Mall = Kanyon
208	Price \geq 4514.9999996928, Shopping Mall = Mall of Istanbul
203	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Istinye Park
201	Price = 1671.2664743936 - 2683.2112533504, Shopping Mall = Mall of Istanbul
199	Price = 1671.2664743936 - 2683.2112533504, Shopping Mall = Kanyon
197	Shopping Mall = Mall of Istanbul, Price = 551.9276546048 - 1671.2664743936

197	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Metropol AVM
191	Shopping Mall = Kanyon, Price = 551.9276546048 - 1671.2664743936
161	Price = 1671.2664743936 - 2683.2112533504, Shopping Mall = Metrocity
158	Shopping Mall = Metrocity, Price = 551.9276546048 - 1671.2664743936
141	Price \geq 4514.9999996928, Shopping Mall = Metrocity
107	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Cevahir AVM
103	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Zorlu Center
102	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Viaport Outlet
101	Price = 1671.2664743936 - 2683.2112533504, Shopping Mall = Istinye Park
99	Price = 2683.2112533504 - 4514.9999996928, Shopping Mall = Emaar Square Mall

Từ thống kê sơ bộ ta có thể đưa ra được đối tượng và chiến lược cho hướng phát triển mảng công nghệ như sau:

Do nhu cầu mua sắm của hầu hết mọi người thường vẫn còn nhắm vào phân khúc giá rẻ nên việc có thêm phân khúc giá rẻ sẽ là một lợi thế để tăng doanh số

Với doanh số bán hàng ở mức doanh thu tầm trung và ở độ tuổi trải dài từ 20-70 hầu hết mức độ quan tâm đến công nghệ của các nhóm tuổi này là như nhau nên ta có thể đẩy mạnh thêm về số lượng mặt hàng và nghiên cứu nhu cầu của người dùng để có thể đem về những mặt hàng công nghệ hợp lý.

Ở mức doanh thu cao ta có thể thấy doanh số của nó không hề nhỏ mà độ tuổi này lại là ở mức 40 trở lên, đây là nhóm độ tuổi đã có nhiều thu nhập và có nhu cầu quan tâm về sức khỏe nên việc thêm các mặt hàng công nghệ liên quan đến chăm sóc sức khỏe ở mức giá cao sẽ góp phần thúc đẩy doanh số và doanh thu cho các Mall.

Điểm cần đặc biệt lưu ý đó là do mọi người vẫn thường lui tới 3 mall lớn là Mall of Istanbul, Kanyon và Metrocitiy nên chúng ta có thể để các mặt hàng công nghệ mới ở đây để xem xét về nhu cầu rồi mới tiến hành phổ biến các mặt hàng này ở các mall còn lại để tránh việc hàng tồn kho với các mặt hàng mắc tiền như mặt hàng công nghệ.

Tương tự với cách khai thác dữ liệu như trên ta sẽ có thể tìm được mong muốn của khách hàng và tiến hành đẩy mạnh lượng hàng hóa và mặt hàng để thúc đẩy tăng doanh thu.

5. Classification

5.1. Decision tree

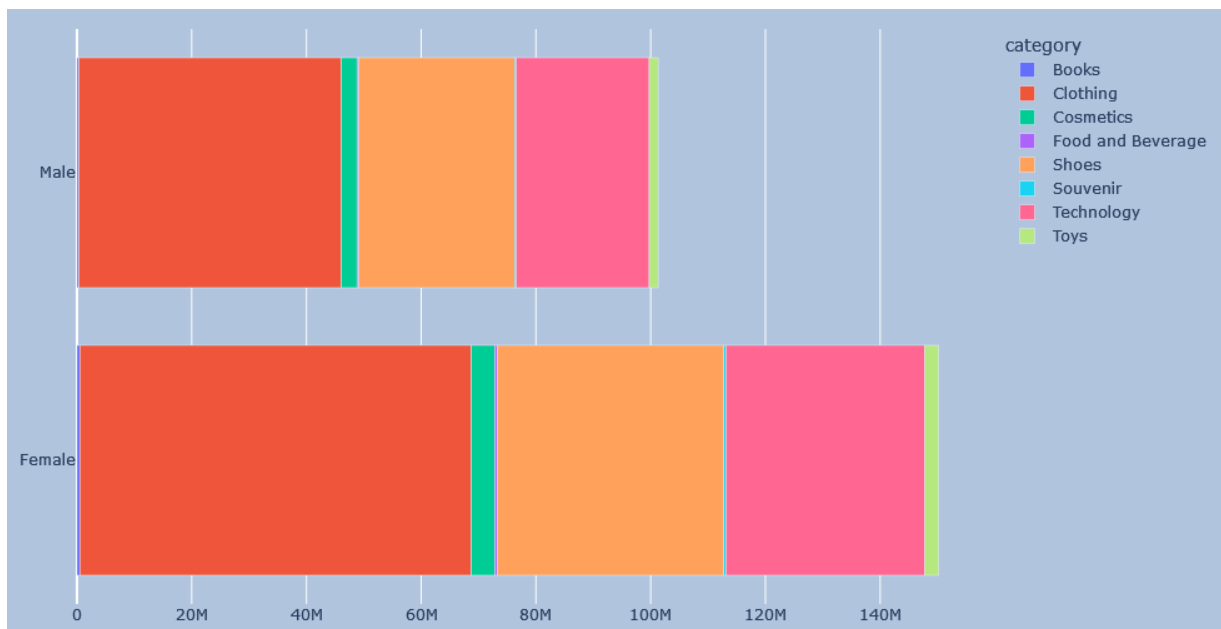
Theo Wikipedia, một cây quyết định (decision tree) là một đồ thị của những quyết định và những hậu quả có thể của nó (bao gồm rủi ro và hao phí tài nguyên). Cây quyết định được sử dụng để xây dựng một kế hoạch nhằm đạt được mục tiêu mong muốn và hỗ trợ quá trình ra quyết định doanh nghiệp.

Đối với use case hiện tại của chúng em, cây quyết định có thể giúp phân loại thông qua quá trình phân nhỏ tập dữ liệu thành những tập con và tạo mô hình cây gồm những nút quyết định (decision node) và nút lá/nút kết quả/nút phân loại (leaf node).

Thông thường, cây quyết định có thể xử lý cả dữ liệu kiểu phân loại và dữ liệu kiểu số. Tuy nhiên, đối với mô hình cây quyết định trong thư viện học máy scikit-learn của Python chúng em sử dụng, tập dữ liệu cần được chuyển hóa thành dữ liệu kiểu số để sử dụng mô hình trên.

5.2. Selecting features

Cây quyết định là mô hình học máy có giám sát nên chúng em cần xác định biến giải thích (feature variable) và biến kết quả (target variable). Để có được một nghiệp vụ có ý nghĩa, chúng em đã trực quan hóa để tìm hiểu về doanh thu của doanh nghiệp dựa trên giới tính khách hàng.



Hình 1. Doanh thu theo khách hàng và danh mục sản phẩm

Với mỗi danh mục sản phẩm, khách hàng nữ đều chi nhiều hơn khi mua sắm. Tuy nhiên, chúng em không thể từ đó và đưa ra quyết định nghiệp vụ marketing hoặc chương trình giảm giá nhắm vào khách hàng nữ. Dựa trên biểu đồ trên, họ là những khách hàng có mức chi tiêu cao hơn khách hàng nam ~20% theo từng danh mục sản phẩm, nhưng đây cũng có thể là vì số lượng khách hàng nữ cao hơn. Khi EDA tập dữ liệu kỹ hơn, chúng em nhận thấy điều này 59.7% khách hàng là nữ, khẳng định suy đoán của chúng em là chính xác. Đây chính là lí do tại sao việc trực quan hóa dữ liệu đã thu thập có thể không chính xác, từ đó, chúng em cần thực hiện áp dụng mô hình học máy cây quyết định để phân loại khách hàng theo giới tính và những thay đổi về chính sách sẽ ảnh hưởng đến khách hàng nào.

Vì vậy, để xác định biến kết quả giới tính (gender), chúng em đã chọn lọc những biến giải thích sau: khung tuổi (age_group), danh mục sản phẩm (category), số lượng (quantity), phương thức thanh toán (payment_method), tổng chi tiêu (total), và lần lượt chia thành tập dữ liệu X và y.

```
features = ['age_group', 'category', 'quantity', 'payment_method', 'total']
targets = ['gender']
X = transactions[features]
y = transactions[targets]
```

Hình 2. Chia tập dữ liệu chứa biến giải thích (X) và biến kết quả (y)

5.3. Transforming data

Như đã nêu trên, mô hình cây quyết định chúng em sử dụng đến từ thư viện scikit-learn chỉ chấp nhận biến giải thích kiểu số và liên tục. Đây cũng là một khuyết điểm của thư viện này, do đối với hệ thống học máy có nền tảng mạnh, cột có kiểu phân loại được xử lý một cách tự nhiên như ngôn ngữ R sẽ sử dụng factors, hoặc Weka sẽ sử dụng kiểu nominal.

Để chuyển đổi kiểu dữ liệu, chúng em có thể thực hiện one-hot-encoding hoặc label-encoding. Điểm khác biệt giữa hai phương pháp trên nằm ở cách mô hình học máy tiếp nhận chúng. Label-encoding hoạt động dựa trên việc gán một giá trị kiểu số nhất định cho một giá trị kiểu phân loại, ví dụ, một cột chứa thống kê về mức chi tiêu gồm ba giá trị riêng biệt “Low”, “Med”, và “Hi” có thể được map sang lần lượt ba giá trị số 1, 2, và 3. Trong trường hợp này, không một cột mới nào được tạo, và mô hình học máy sẽ xem cột vừa được áp dụng label-encoding có thứ tự hoặc cấp bậc, do $1 < 2 < 3$.

Trong những biến giải thích, cột độ tuổi (age_group) phù hợp với phương pháp vừa nêu trên. Đối với những biến giải thích kiểu phân loại còn lại (danh mục sản phẩm và phương thức thanh toán), chúng em sẽ áp dụng one-hot-encoding. Phương pháp này sẽ tạo n cột kiểu boolean theo n giá trị riêng biệt của cột ban đầu.

5.4. Splitting train and test data

Để xác định độ chính xác của mô hình, nhóm sẽ chia tập dữ liệu thành hai phần: train và test với lần lượt 70% và 30% số lượng giao dịch từ tập dữ liệu gốc.


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1, stratify=y)
```

Hình 3. Chia tập dữ liệu

Tham số `random_state=1` sẽ đảm bảo mỗi lần chạy mô hình, phân tách sẽ luôn giống nhau để kết quả mô hình có thể tái tạo lại (reproducible). Với `stratify=y`, chúng em sẽ loại bỏ được hoặc ít nhất giảm thiểu những trường hợp thiên vị dữ liệu giữa tập train và tập test, như tập train chỉ chứa toàn khách hàng nữ, ảnh hưởng đến kết quả phân loại.

5.5. Build model

Sau khi fit tập dữ liệu train vào mô hình cây quyết định của thư viện `scikit-learn`, độ chính xác của mô hình là ~ 0.5924 , hoặc nói cách khác, mô hình cây quyết định có thể phân loại đúng 59,24% trên tổng số trường hợp. Tuy con số này không hẳn là tệ, mô hình của chúng em gặp vấn đề về overfitting.

Overfitting là một hành vi mà chúng em không muốn gặp phải khi xây dựng mô hình học máy. Những mô hình gặp vấn đề overfitting sẽ đưa kết quả dự đoán (phân loại) chính xác đối với dữ liệu được học, nhưng không thể đảm bảo độ chính xác khi sử dụng chính mô hình trên với dữ liệu mới. Trong trường hợp của chúng em, vấn đề overfitting nằm ở việc chúng em chưa áp đặt những hyperparameter của mô hình học máy một cách hợp lý như cắt (pruning) hoặc đặt giới hạn chiều sâu của cây (`max_depth`), do đó, cây quyết định hiện tại rất to, không mang lại ý nghĩa nghiệp vụ, và tốn thời gian phân tích vì độ phức tạp cao. Một lí do có thể kể đến là tập dữ liệu train chứa quá nhiều cột, khiến số lượng nút quyết định tăng mạnh.

5.6. Improving accuracy

Hyperparameters là những tham số có thể được định nghĩa lúc xây dựng mô hình học máy. Với cây quyết định, việc cấu hình quy luật thuật toán sử dụng để phân quyết định (theo entropy hoặc gini) hoặc chiều sâu tối đa của cây có thể giúp tăng độ chính xác của mô hình và quan trọng nhất, tránh overfitting.

Chúng em sẽ tự động hóa việc tìm ra tổ hợp tham số tốt nhất cho mô hình cây quyết định sử dụng GridSearchCV, phương pháp xác thực chéo (cross-validation). Do chúng em không thể xác định được tổ hợp tham số tốt nhất, chúng em cần thử nhiều tập khác nhau để so sánh độ chính xác giữa những lần chạy mô hình và biết được tập nào là tối ưu nhất. Nếu thực hiện một cách thủ công, việc này có thể tốn hàng giờ, thậm chí ngày và tuần, nên GridSearchCV là phương pháp tối ưu nhất cho việc điều chỉnh hyperparameters.

Tham số chúng em muốn thay đổi là:

- **criterion**: cách mô hình cây quyết định chia theo gini hoặc entropy
- **max_depth**: chiều sâu tối đa của cây
- **max_features**: số lượng biến giải thích được xem xét khi tìm kiếm phương án phân chia tốt nhất
- **splitter**: phương pháp tìm kiếm biến giải thích để thực hiện phân chia theo best hoặc random

```
params = {
    'criterion': ['gini', 'entropy'],
    'max_depth': [None, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'max_features': [None, 'sqrt', 'log2', 0.2, 0.4, 0.6, 0.8] + list(range(1, 10)),
    'splitter': ['best', 'random']
}

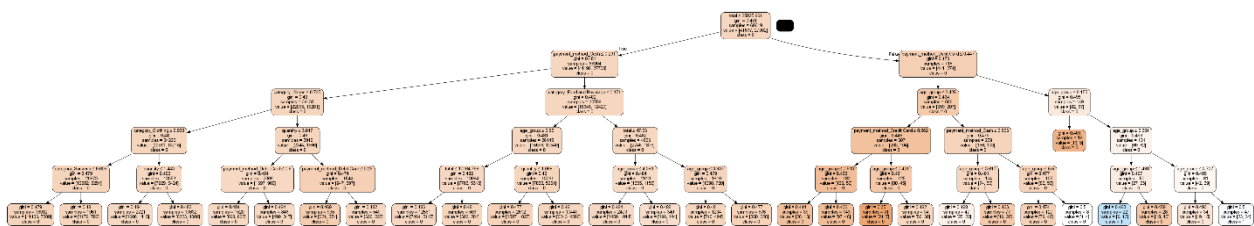
clf = GridSearchCV(estimator=DecisionTreeClassifier(), param_grid=params, cv=5, n_jobs=-1, verbose=1)
clf.fit(X_train, y_train)
clf.best_params_
```

Fitting 5 folds for each of 704 candidates, totalling 3520 fits

```
{'criterion': 'gini',
 'max_depth': 5,
 'max_features': 0.2,
 'splitter': 'random'}
```

Hình 4. Sử dụng GridSearchCV

Với tổ hợp tham số tối ưu nhất (lúc chạy notebook bảy giờ), độ chính xác của mô hình tăng lên ~0.5982. Tuy độ chính xác không tăng nhiều, mô hình học máy đã không còn gặp vấn đề overfitting.



Hình 5. Cây quyết định

Với mỗi lần chạy notebook, GridSearchCV sẽ đưa ra một tổ hợp tham số tối ưu mới. Trong mô hình trên, mỗi ô là một nút cho biết quy luật phân chia và những nút lá cho biết biến kết quả phân loại cuối cùng. Mô hình này có thể giúp trả lời rất nhiều câu hỏi nghiệp vụ về giới tính, như nếu khách hàng đã chi hơn 25825, không sử dụng phương thức thẻ ghi nợ, và là khách hàng trẻ trong độ tuổi 18-24, thì giới tính là nữ (đi theo nhánh bên phải), từ đó, việc quyết định thay đổi chính sách thanh toán bằng thẻ ghi nợ có thể ảnh hưởng khách hàng nữ trong độ tuổi trên.