ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

KHOA TOÁN - TIN
Faculty of Mathematics and Informatics

Mô hình luật kết hợp Customer Shopping

Hệ hỗ trợ quyết định

Giảng viên hướng dẫn: TS. Lê Hải Hà

Nguyễn Hồng Ánh 20216798
Hoàng Thị Ngân 20216860

Nhóm thực hiện: Nhóm 5 - Lớp 150330 - Học kỳ 2023.2

Ngày 24 tháng 6 năm 2024

Mục lục

Mở đầu

Trong thời đại số hóa ngày nay, khối lượng dữ liệu được sinh ra từ các nguồn khác nhau đã trở nên vô cùng phong phú và phức tạp. Để khai thác tối đa giá trị từ những dữ liệu này, nhiều nhà nghiên cứu và chuyên gia đã đặt nỗ lực vào việc tìm ra những phương pháp tiên tiến để phân tích và hiểu được mô hình, mẫu chẩn và quy tắc bên trong dữ liệu. Trong lĩnh vực này, công nghệ khai phá luật kết hợp đã nổi lên như một lĩnh vực quan trọng và mở ra cánh cửa mới cho việc khám phá tri thức ẩn trong dữ liệu.

Báo cáo của chúng em bao gồm năm chương:

- Chương 1 giới thiệu về đề tài và phát biểu bài toán.
- Chương 2 phân tích và tiền xử lý dữ liệu.
- Chương 3 xây dựng mô hình với thuật toán Apriori và FP-Growth.
- Chương 4 ứng dụng của mô hình trong bài toán gợi ý phim với dữ liệu về lịch sử xem phim anime của người dùng.
- Chương 5 kết luận về các cách tiếp cận đề tài và khả năng ứng dụng của kết quả nghiên cứu trong tương lai.

Chúng em xin gửi lời cảm ơn sâu sắc nhất tới TS. Lê Hải Hà, giảng viên môn Hệ hỗ trợ quyết định đã tận tâm giảng dạy, hướng dẫn, giúp đỡ để em có thể hoàn thành báo cáo môn học này.

1 Phát biểu bài toán

1.1 Cơ sở lý thuyết

Association Rule (luật kết hợp) là một quy trình phân tích dữ liệu nhằm tìm ra những mẫu tương quan và quy tắc kết hợp giữa các mục trong tập dữ liệu.

Mục tiêu của lớp thuật toán này không phải để dự đoán sự xuất hiện của 1 mục khác mà là tìm các mẫu có thể xuất hiện cùng lúc với các mẫu khác.

Việc học các quy tắc kết hợp là 1 nhánh của quá trình học không giám sát, khám phá các mẫu ẩn trong dữ liệu.

Một trong những ứng dụng phổ biến của kĩ thuật này được gọi là *phân tích xu hướng người dùng* để tìm ra sự xuất hiện của các thuộc tính, xu hướng này với các thuộc tính, xu hướng khác trong cùng một giao dịch (transaction).

Luật kết hợp có dạng : X $\Rightarrow$ Y với X, Y $\subset$ I (I là tập các hạng mục (itemset)) và X $\cap$ Y = $\emptyset$.

Ý nghĩa: Khi X có mặt thì Y xuất hiện với một xác suất nào đó.

Trong đề tài này, ta chú ý đến 3 thông số quan trọng đó là :

- support (độ hỗ trợ): thể hiện tần suất xuất hiện cả X và Y trong tổng số giao dịch.

$$\text{support}(X \cup Y) = \frac{\text{Số giao dịch chứa X và Y}}{\text{Tổng số giao dịch}}$$

- confidence (độ tin cậy): đo lường mức độ tin tưởng vào một luật cụ thể. Độ tin cậy cao cho thấy mức độ chắc chắn của luật.

$$\text{confidence}(X \longrightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X)}$$

- lift: đo lường mức độ mà sự xuất hiện của một mục (item) trong luật kết hợp ảnh hưởng đến sự xuất hiện của mục khác, so với khi chúng độc lập với nhau.

$$\text{lift}(X \longrightarrow Y) = \frac{\text{support}(X \cup Y)}{\text{support}(X) \times \text{support}(Y)}$$

1.2 Mô tả bài toán

Bài toán

Một cửa hàng bán lẻ muốn phân tích thông tin về các đơn hàng của họ để tìm ra các quy luật và xu hướng mua sắm của khách hàng. Bộ dữ liệu mà cửa hàng đã tổng hợp lại, gọi là *Customer Shopping Trends Dataset*, chứa thông tin về các đơn hàng bao gồm các thông tin như thông tin khách hàng (tuổi, giới tính, địa điểm), tên sản phẩm, giá sản phẩm, thời điểm, phương thức thanh toán,..... Thông qua bộ dữ liệu này, cửa hàng muốn hiểu rõ hơn về hành vi mua sắm của khách hàng nhằm điều chỉnh chiến lược tiếp thị và cải thiện trải nghiệm tổng thể của khách hàng.

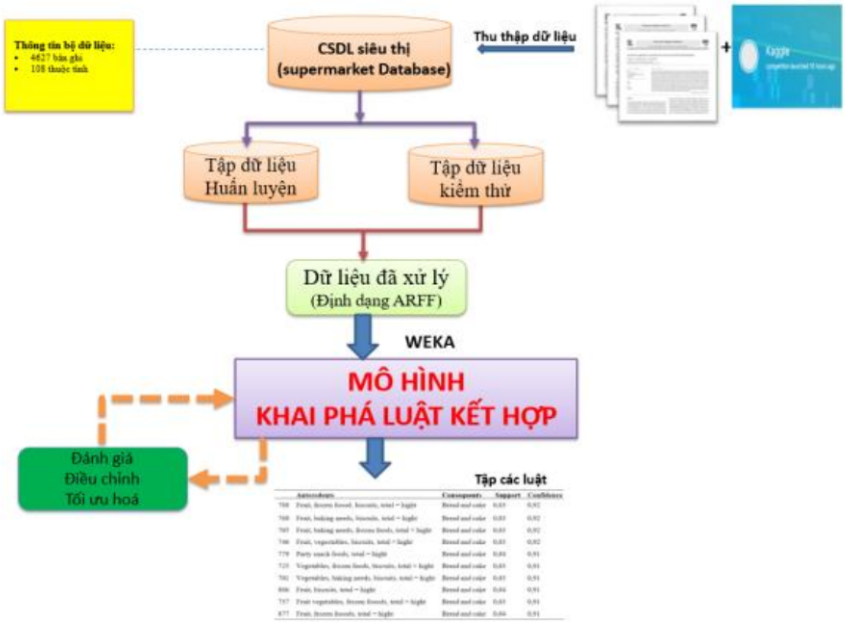
Yêu cầu xử lý:

- Đọc và truy xuất dữ liệu

- Phân tích dữ liệu để tìm ra các mô hình và quy luật kết hợp giữa các thuộc tính của một hóa đơn mua hàng
- Áp dụng thuật toán để tìm ra các tập đối tượng hay đi cùng nhau, từ đó rút ra các quy luật kết hợp
- Diễn giải kết quả và ý nghĩa

Input: Bộ dữ liệu là hành vi mua sắm của khách hàng bao gồm nhiều thuộc tính của khách hàng như độ tuổi, giới tính, lịch sử mua hàng, phương thức thanh toán ưa thích, tần suất mua hàng, v.v.

Output: Các quy luật có giá trị về xu hướng của khách hàng, giúp doanh nghiệp điều chỉnh chiến lược tiếp thị và cải thiện trải nghiệm tổng thể của khách hàng.



Hình 1: Mô hình khai phá luật kết hợp

2 Phân tích và tiền xử lý dữ liệu

2.1 Giới thiệu về bộ dữ liệu

2.1.1 Thu thập dữ liệu

Bộ dữ liệu mà chúng em sử dụng trong báo cáo này có tên là : *"Customer Shopping Trends Dataset"* được thu thập từ *kaggle.com* Link database.

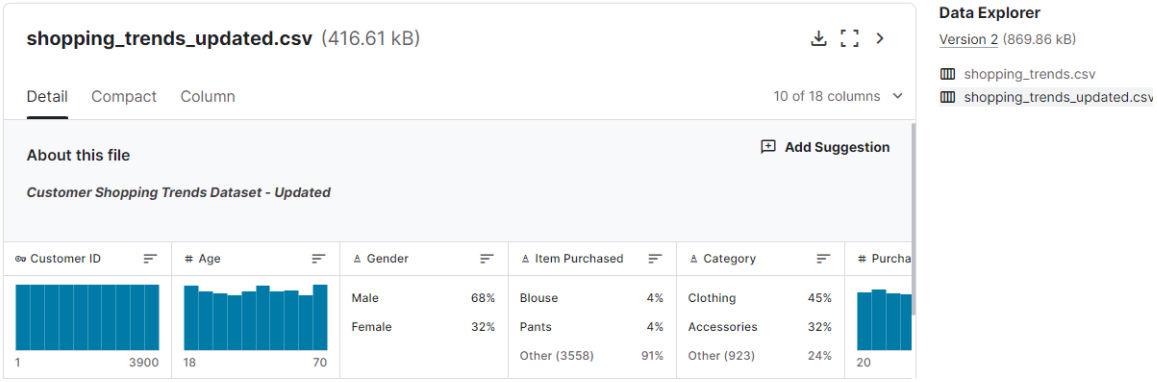
```
df = pd.read_csv("/kaggle/input/customer-shopping-trends-dataset/
    shopping_trends_updated.csv")
```

2.1.2 Thông kê dữ liệu mẫu

a. Quy mô bộ dữ liệu như sau :

- Kích thước bộ dữ liệu: 869.86 KB.

- Định dạng file dữ liệu: csv.
- Số file dữ liệu: 2.
- Số trường dữ liệu của file *shopping_trends_updated*: 18. Còn của file *shopping_trends*: 19.



Ở đây bọn em sẽ sử dụng chủ yếu file dữ liệu *shopping_trends_updated.csv*.

b. Chi tiết bảng dữ liệu shopping_trends_updated.csv

- Customer ID: Mã định danh duy nhất của khách hàng.
- Age: Tuổi của khách hàng.
- Gender: Giới tính của khách hàng (Male/Female).
- Item Purchased: Mặt hàng mà khách hàng mua.
- Category: Danh mục mặt hàng.
- Purchase Amount (USD): Số tiền khách hàng trả cho đơn hàng.
- Location: Địa điểm mua hàng.
- Size: Kích thước mặt hàng.
- Color: Màu sắc của mặt hàng đã mua.
- Season: Mùa trong việc mua hàng.
- Review Rating: Đánh giá mặt hàng đã mua do khách hàng đưa ra.
- Subscription Status: Trạng thái đăng ký của khách hàng.
- Shipping Type: Loại vận chuyển được lựa chọn.
- Discount Applied: Giảm giá được áp dụng hay không (Yes/No).
- Promo Code Used: Mã khuyến mãi có được sử dụng không (Yes/No).
- Previous Purchases: Tổng số giao dịch được khách hàng thực hiện tại cửa hàng, không bao gồm giao dịch đang diễn ra.
- Payment Method: Phương thức thanh toán ưa thích nhất của khách hàng.

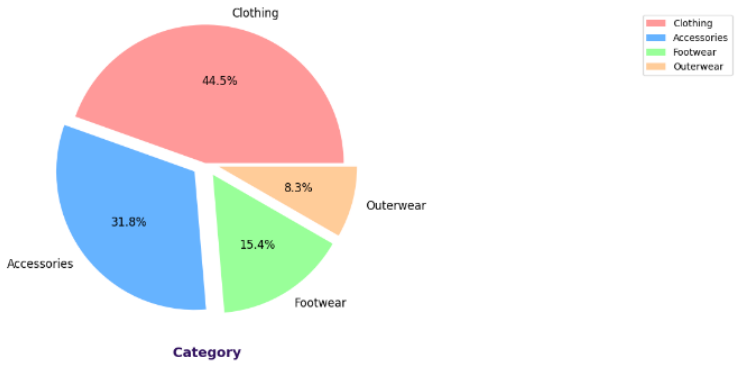
- Frequency of Purchases: Tần suất mua hàng của khách hàng.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer ID             3900 non-null   int64
 1   Age                     3900 non-null   int64
 2   Gender                  3900 non-null   object
 3   Item Purchased          3900 non-null   object
 4   Category                3900 non-null   object
 5   Purchase Amount (USD)   3900 non-null   int64
 6   Location                3900 non-null   object
 7   Size                    3900 non-null   object
 8   Color                   3900 non-null   object
 9   Season                  3900 non-null   object
 10  Review Rating           3900 non-null   float64
 11  Subscription Status     3900 non-null   object
 12  Shipping Type           3900 non-null   object
 13  Discount Applied        3900 non-null   object
 14  Promo Code Used         3900 non-null   object
 15  Previous Purchases      3900 non-null   int64
 16  Payment Method          3900 non-null   object
 17  Frequency of Purchases  3900 non-null   object
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

Hình 2: Số dòng dữ liệu và định dạng dữ liệu trong các cột.

2.2 Data exploration

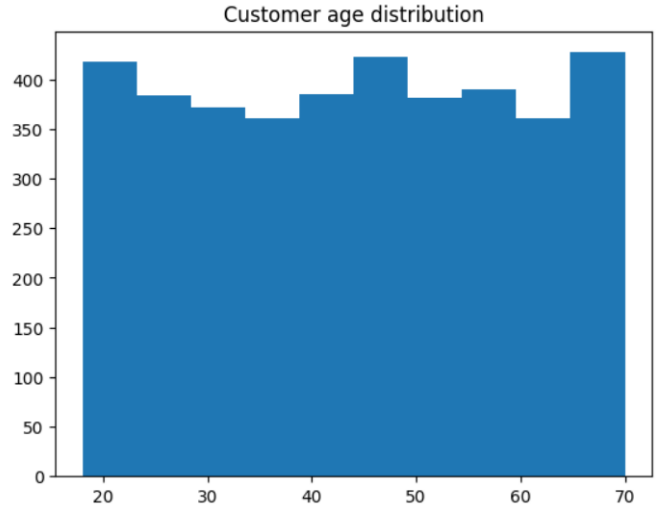
1. Biểu đồ tỷ lệ dannh mục sản phẩm



- Tỷ lệ chiếm lĩnh: Clothing là danh mục chiếm tỷ trọng lớn nhất, cho thấy có sự tập trung mạnh vào các sản phẩm quần áo.
- Phân bổ danh mục: Các danh mục khác như phụ kiện và Acesories cũng chiếm một phần đáng kể, cho thấy sự đa dạng hóa sản phẩm.
- Trang phục ngoài trời có tỷ lệ nhỏ nhất, có thể là cơ hội để mở rộng hoặc phát triển thêm sản phẩm trong danh mục này nếu thị trường có nhu cầu.

=> Cho thấy một bức tranh tổng quan về sự phân bố sản phẩm trong các danh mục, giúp doanh nghiệp nhận diện được đâu là danh mục chính và đâu là danh mục cần có sự chú trọng hoặc phát triển thêm.

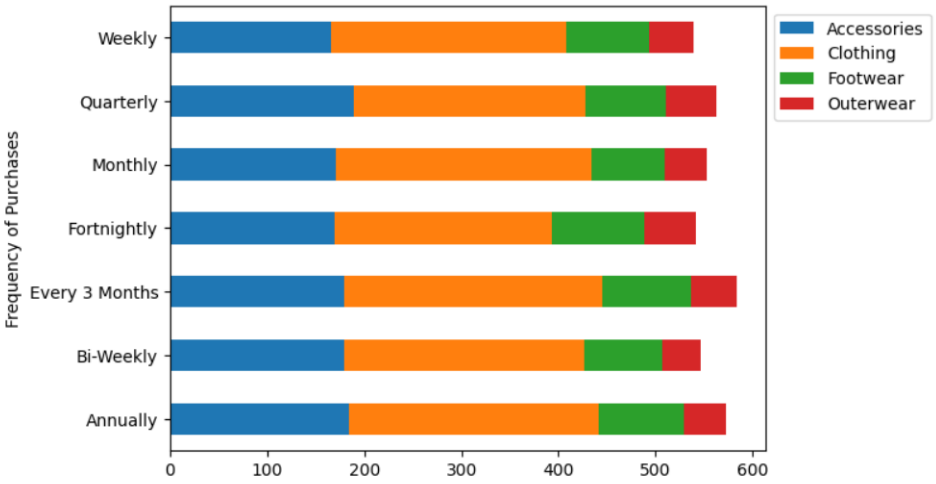
2. Biểu đồ số lượng khách hàng theo tuổi.



- Khoảng tuổi: Độ tuổi của khách hàng dao động từ khoảng 20 đến 70 tuổi.
- Biểu đồ cho thấy khách hàng phân bố khá đồng đều qua các nhóm tuổi khác nhau.
- Có một vài đỉnh điểm nhẹ xung quanh các độ tuổi 20, 40, và 70, mỗi đỉnh có khoảng 400 khách hàng.

=>Doanh nghiệp hoặc dịch vụ dường như thu hút một dải rộng các nhóm tuổi, cho thấy sự hấp dẫn rộng rãi đối với nhiều nhóm độ tuổi khác nhau.
Không có sự thiên vị đáng kể về độ tuổi, cho thấy tiềm năng cho các chiến dịch tiếp thị hoặc phát triển sản phẩm nhắm đến tất cả các nhóm tuổi.

3. Biểu đồ thanh ngang xếp chồng thể hiện tần suất mua hàng theo các loại mặt hàng khác nhau

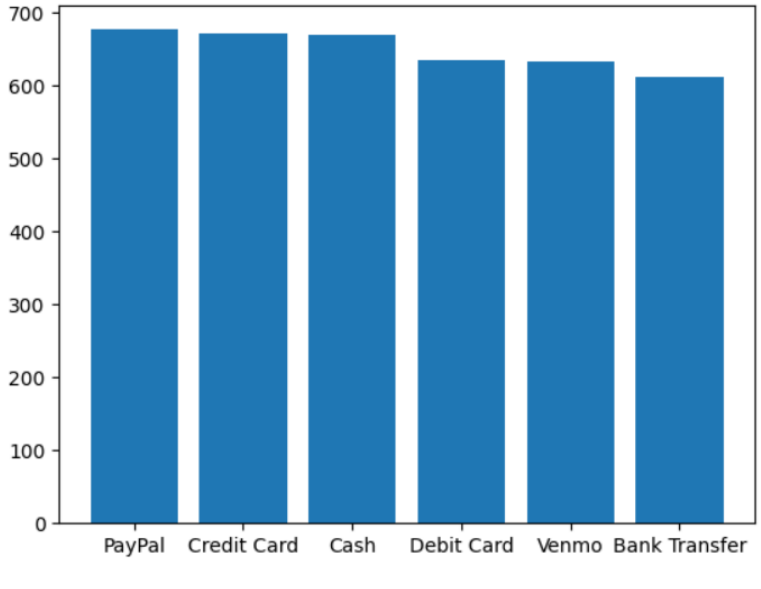


- Weekly and Bi-Weekly Purchases: accessories và clothing chiếm ưu thế.
- Monthly, Quarterly, and Every 3 Months Purchases: Cân bằng hơn giữa clothing, footwear và outerwear, accessories chiếm thấp hơn.

- Annual Purchases: Clothing là mặt hàng được mua nhiều nhất, tiếp theo là footwear và outerwear, với phụ kiện ít được mua nhất.

Biểu đồ này cho thấy các mặt hàng thiết yếu hoặc cần thiết thường xuyên như accessories và clothing được mua nhiều hơn, trong khi các mặt hàng có chi phí cao hơn hoặc cần ít thường xuyên hơn như footwear và outerwear ngoài được mua ít thường xuyên hơn.

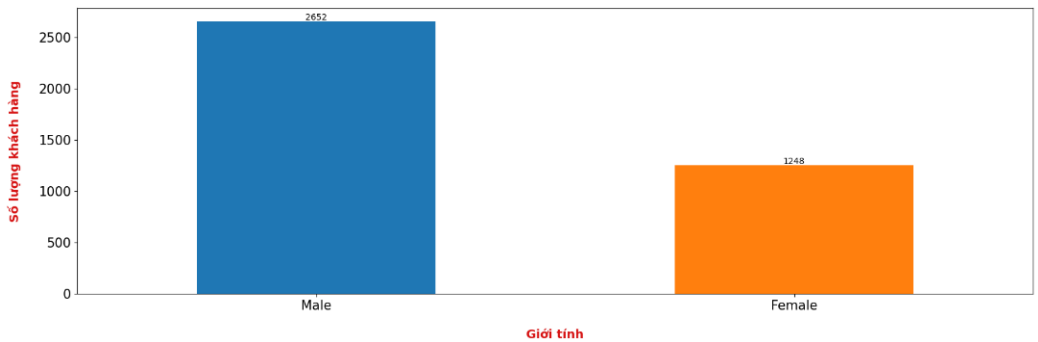
4. Biểu đồ thống kê số lượng khách hàng sử dụng các phương thức thanh toán khác nhau.



- PayPal và Thẻ tín dụng là hai phương thức thanh toán phổ biến nhất với số lượng sử dụng gần như ngang bằng, đạt khoảng 700 lượt.
- Tiền mặt cũng được sử dụng khá nhiều, nhưng ít hơn một chút so với PayPal và Thẻ tín dụng.
- Thẻ ghi nợ và Venmo có số lượng sử dụng thấp hơn một chút so với Tiền mặt, nhưng vẫn ở mức tương đối cao.
- Chuyển khoản ngân hàng là phương thức ít được sử dụng nhất trong số các phương thức thanh toán được khảo sát, nhưng không chênh lệch quá lớn so với các phương thức khác.

=> Cải thiện giao diện và trải nghiệm người dùng khi sử dụng các phương thức này có thể dẫn đến sự hài lòng cao hơn và tăng tỷ lệ hoàn tất đơn hàng.

5. Biểu đồ thể hiện số lượng khách hàng theo Gender.



- Số lượng khách hàng Nam (2652) gần gấp đôi số lượng khách hàng Nữ (1248). Điều này cho thấy có sự chênh lệch đáng kể về số lượng khách hàng giữa hai giới tính.

=> Cho thấy sự chênh lệch rõ rệt về số lượng khách hàng giữa Nam và Nữ. Doanh nghiệp nên cân nhắc điều chỉnh chiến lược kinh doanh dựa trên dữ liệu này để phục vụ tốt hơn cho nhóm khách hàng chiếm số lượng lớn cũng như thu hút thêm nhóm khách hàng còn lại.

2.3 Tiền xử lý dữ liệu

Xử lý giá trị ngoại lệ

Thực hiện kiểm tra xem trong bộ dữ liệu có giá trị Null không thông qua đoạn mã :

```
df.isnull().sum()
```

Kết quả trả ra là : 0 có trường dữ liệu trống

```
[27]: Customer ID                0
      Age                        0
      Gender                     0
      Item Purchased             0
      Category                   0
      Purchase Amount (USD)      0
      Location                   0
      Size                       0
      Color                      0
      Season                     0
      Review Rating              0
      Subscription Status        0
      Shipping Type              0
      Discount Applied           0
      Promo Code Used            0
      Previous Purchases         0
      Payment Method             0
      Frequency of Purchases     0
      dtype: int64
```

Lọc theo mùa

Vì hiện tại là mùa hè nên chúng em sẽ tập trung phân tích xu hướng vào mùa hè.

```
df = df[df['Season'] == 'Summer']
```

Lấy các thuộc tính cần thiết

Ở đây, chúng em tập trung phân tích phương thức thanh toán và tần suất mua hàng để giúp doanh nghiệp hiểu rõ các phân khúc khách hàng thích sử dụng hình thức thanh toán nào nhất, từ đó có thể cung cấp thêm các tùy chọn thanh toán tiện lợi hơn đồng thời có thể đưa ra các chương trình khuyến mãi theo tần suất mua của khách hàng và hợp tác với các ngân hàng, app thanh toán điện tử để tạo ra các chương trình khuyến mãi, tích điểm nhằm khuyến khích khách hàng mua sắm.

```
df = df[['Age','Frequency of Purchases','Payment Method']]
```

	Age	Frequency of Purchases	Payment Method
5	46	Weekly	Venmo
8	26	Annually	Venmo
18	52	Weekly	Cash
19	66	Bi-Weekly	Debit Card
22	56	Annually	Debit Card
...	...	...	...
3878	60	Annually	Credit Card
3886	37	Quarterly	Debit Card
3892	35	Fortnightly	PayPal
3895	40	Weekly	Venmo
3898	44	Weekly	Venmo

955 rows × 3 columns

Binning dữ liệu

Biến đổi dữ liệu về tuổi thành các khoảng.

```
age_bins = [18, 30, 40, 50, 60, 70]
age_labels = ['18-29', '30-39', '40-49', '50-59', '60-70']
df['Age'] = pd.cut(df['Age'], age_bins, labels = age_labels,
    include_lowest = True)
```

	Age	Frequency of Purchases	Payment Method
5	40-49	Weekly	Venmo
8	18-29	Annually	Venmo
18	50-59	Weekly	Cash
19	60-70	Bi-Weekly	Debit Card
22	50-59	Annually	Debit Card
...	...	...	...
3878	50-59	Annually	Credit Card
3886	30-39	Quarterly	Debit Card
3892	30-39	Fortnightly	PayPal
3895	30-39	Weekly	Venmo
3898	40-49	Weekly	Venmo

Mã hóa one-hot

Thêm tên thuộc tính vào mỗi giá trị

```
df = pd.DataFrame({col:str(col) + '=' for col in df}, index = df.index)
    + df.astype(str)
```

	Age	Frequency of Purchases	Payment Method
5	Age=40-49	Frequency of Purchases=Weekly	Payment Method=Venmo
8	Age=18-29	Frequency of Purchases=Annually	Payment Method=Venmo
18	Age=50-59	Frequency of Purchases=Weekly	Payment Method=Cash
19	Age=60-70	Frequency of Purchases=Bi-Weekly	Payment Method=Debit Card
22	Age=50-59	Frequency of Purchases=Annually	Payment Method=Debit Card
...	...	...	...
3878	Age=50-59	Frequency of Purchases=Annually	Payment Method=Credit Card
3886	Age=30-39	Frequency of Purchases=Quarterly	Payment Method=Debit Card
3892	Age=30-39	Frequency of Purchases=Fortnightly	Payment Method=PayPal
3895	Age=30-39	Frequency of Purchases=Weekly	Payment Method=Venmo
3898	Age=40-49	Frequency of Purchases=Weekly	Payment Method=Venmo

Tạo ra danh sách tất cả các giá trị trong bảng dữ liệu

```
df_list = df.to_numpy().tolist()
df_list
dataset = list()
for i in range(len(df_list)):
    dataset.append(df_list[i])
```

Mã hóa one-hot

```
from mlxtend.preprocessing import TransactionEncoder
te = TransactionEncoder()
te_array = te.fit(dataset).transform(dataset)
final_df = pd.DataFrame(te_array, columns=te.columns_)
```

	Age=18-29	Age=30-39	Age=40-49	Age=50-59	Age=60-70	Frequency of Purchases=Annually	Frequency of Purchases=Bi-Weekly	Frequency of Purchases=Every 3 Months	Frequency of Purchases=Fortnightly	Frequency of Purchases=Monthly	Frequency Purchases=Quarte
0	False	False	True	False	False	False	False	False	False	False	Fa
1	True	False	False	False	False	True	False	False	False	False	Fa
2	False	False	False	True	False	False	False	False	False	False	Fa
3	False	False	False	False	True	False	True	False	False	False	Fa
4	False	False	False	True	False	True	False	False	False	False	Fa
...	...	...	...	...	...	...	...	...	...	...	
950	False	False	False	True	False	True	False	False	False	False	Fa
951	False	True	False	False	False	False	False	False	False	False	Tı
952	False	True	False	False	False	False	False	False	True	False	Fa
953	False	True	False	False	False	False	False	False	False	False	Fa
954	False	False	True	False	False	False	False	False	False	False	Fa

955 rows × 18 columns

3 Xây dựng mô hình và đánh giá

3.1 Xây dựng luật kết hợp với giải thuật Apriori

3.1.1 Giải thuật Apriori

1. Ý tưởng thuật toán

Dựa trên tính chất Apriori của tập phổ biến.
- Tìm tất cả các tập phổ biến 1-hạng mục. Sau đó là tất cả các tập phổ biến, 2-hạng mục,...
- Trong mỗi vòng lặp k, chỉ quan tâm đến những tập có chứa một số các tập phổ biến (k-1)- hạng mục.
- Tạo các tập ứng viên kích thước k-hạng mục từ các tập phổ biến có kích thước (k-1)-hạng mục.
- Kiểm tra độ phổ biến của các ứng viên CSDL và loại các không phổ biến.

2. Đặc điểm của thuật toán

- Tạo ra nhiều tập ứng viên. Ví dụ :10^4 frequent 1-itemsets nhiều hơn 10^7 2-itemsets ứng viên.
- Một k-itemset cần ít nhất 2k -1 itemsets ứng viên trước đó.
- Kiểm tra tập dữ liệu nhiều lần. - Chi phí lớn khi kích thước các itemsets tăng lên dần. Nếu k-itemsets được khám phá thì cần kiểm tra tập dữ liệu k+1 lần. Hiện nay, người ta đã cải tiến giải thuật Apriori.

- Kỹ thuật dựa trên bảng băm (hash-based technique). Một k-itemset ứng với hashing bucket count nhỏ hơn minimum support threshold không là một frequent itemset.
- Giảm giao dịch (transaction reduction): Một giao dịch không chứa frequent k-itemset nào thì không cần được kiểm tra ở các lần sau (cho k+1-itemset).
- Phân hoạch (partitioning): Một itemset phải frequent trong ít nhất một phân hoạch thì mới có thể frequent trong toàn bộ tập dữ liệu.
- Lấy mẫu (sampling): Khai phá chỉ tập con dữ liệu cho trước với một trị support threshold nhỏ hơn và cần một phương pháp để xác định tính toàn diện (completeness).
- Đếm itemset động (dynamic itemset counting): Chỉ thêm các itemsets ứng tuyển khi tất cả các tập con của chúng được dự đoán là frequent.

3. Thực hiện code giải thuật Apriori cải tiến.

Trước hết, chúng ta sẽ xây dựng phương thức 'generate_candidates()' để sinh ra các tập candidate itemset có k phần tử

```
from itertools import combinations

def generate_candidates(prev_candidates, k):
    candidates = []
    n_candidates = len(prev_candidates)

    for i in range(n_candidates):
        for j in range(i + 1, n_candidates):
            itemset1 = list(sorted(prev_candidates[i]))
            itemset2 = list(sorted(prev_candidates[j]))

```

```
            if itemset1[:k-2] != itemset2[:k-2]:
                continue

            new_candidate = set(itemset1).union(set(itemset2))
            is_valid_candidate = True
            for subset in combinations(new_candidate, k - 1):
                if frozenset(subset) not in prev_candidates:
                    is_valid_candidate = False
                    break

            if is_valid_candidate is True:
                candidates.append(new_candidate)
    return candidates
```

Sau đó chúng ta sẽ xây dựng phương thức 'apriori_algorithm()' để tìm ra các quy luật của các transaction

```
def apriori(transactions, min_support):
    itemsets = {}
    n = len(transactions)
    min_support_count = n * min_support

    itemsets[1] = []
    item_counts = {}

    for transaction in transactions:
        for item in transaction:
            if item in item_counts:
                item_counts[item] += 1
            else:
                item_counts[item] = 1

    for item, count in item_counts.items():
        if count >= min_support_count:
            itemsets[1].append(frozenset((item,)))

    k = 2
    while itemsets[k-1]:

        candidates = generate_candidates(itemsets[k-1], k)


        candidate_counts = {candidate: 0 for candidate in candidates}

        for transaction in transactions:
            for candidate in candidates:
                if candidate.issubset(transaction):
                    candidate_counts[candidate] += 1
        itemsets[k] = []
        for candidate, count in candidate_counts.items():
            if count >= min_support_count:
                itemsets[k].append(candidate)
        k += 1

```

```
38     return itemsets
```

3.1.2 Mô tả điều kiện dừng của thuật toán

Thuật toán Apriori là một trong những thuật toán phổ biến nhất dùng để khai thác tập mục thường xuyên (frequent itemsets) trong cơ sở dữ liệu giao dịch. Điều kiện dừng của thuật toán Apriori được xác định khi không còn có thể tìm thấy tập mục thường xuyên mới từ cơ sở dữ liệu giao dịch nữa.

3.1.3 Tạo mô hình với bộ dữ liệu : Customer Shopping Trends Dataset

Để giải quyết bài toán, ta sẽ sử dụng giải thuật Apriori để tạo mô hình luật kết hợp với các thuộc tính nhằm xác định xu hướng của khách hàng.
Ta sẽ sử dụng thuật toán Apriori để đưa ra các tập phổ biến với min_support = 0.02.

```
1 frequent_itemsets = apriori(final_df, min_support=0.02, use_colnames=
    True)
```

	support	itemsets
0	0.253403	(Age=18-29)
1	0.178010	(Age=30-39)
2	0.193717	(Age=40-49)
3	0.195812	(Age=50-59)
4	0.179058	(Age=60-70)
...	...	...
108	0.026178	(Frequency of Purchases=Quarterly, Payment Met...
109	0.021990	(Frequency of Purchases=Weekly, Payment Method...
110	0.028272	(Frequency of Purchases=Weekly, Payment Method...
111	0.030366	(Frequency of Purchases=Weekly, Payment Method...
112	0.020942	(Frequency of Purchases=Weekly, Payment Method...

113 rows × 2 columns

Các quy luật rút ra với lift min = 1 theo thứ tự giảm dần confidence và có chứa giá trị tuổi trong tiên đề.

```
1 rules = association_rules(frequent_itemsets, metric="lift",
    min_threshold=1)
2 rules = rules.sort_values(['confidence', 'lift'], ascending =[False,
    False])
3 specific_word = 'Age'
4 filtered_rules = rules[rules['antecedents'].apply(lambda x:  any(
    specific_word in item for item in x))]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
26	(Frequency of Purchases=Monthly, Age=50-59)	(Payment Method=PayPal)	0.023037	0.167539	0.009424	0.409091	2.441761	0.005565	1.408780	0.604383
33	(Frequency of Purchases=Weekly, Age=50-59)	(Payment Method=Bank Transfer)	0.026178	0.167539	0.009424	0.360000	2.148750	0.005038	1.300720	0.548984
15	(Frequency of Purchases=Fortnightly, Age=30-39)	(Payment Method=Cash)	0.027225	0.169634	0.009424	0.346154	2.040598	0.004806	1.269972	0.524220
1	(Frequency of Purchases=Monthly, Age=18-29)	(Payment Method=Venmo)	0.032461	0.164398	0.010471	0.322581	1.962194	0.005135	1.233508	0.506818
44	(Frequency of Purchases=Weekly, Age=60-70)	(Payment Method=Cash)	0.029319	0.169634	0.009424	0.321429	1.894841	0.004451	1.223698	0.486516
16	(Payment Method=Cash, Age=30-39)	(Frequency of Purchases=Fortnightly)	0.030366	0.132984	0.009424	0.310345	2.333695	0.005386	1.257173	0.589393
40	(Age=60-70, Payment Method=Debit Card)	(Frequency of Purchases=Quarterly)	0.030366	0.150785	0.009424	0.310345	2.058190	0.004845	1.231361	0.530238
21	(Payment Method=Venmo, Age=40-49)	(Frequency of Purchases=Every 3 Months)	0.035602	0.159162	0.010471	0.294118	1.847910	0.004805	1.191187	0.475787
38	(Frequency of Purchases=Quarterly, Age=60-70)	(Payment Method=Debit Card)	0.032461	0.165445	0.009424	0.290323	1.754798	0.004054	1.175964	0.444565

Chỉnh min_threshold của lift lên 1.5 để quan sát nhưng luật có sự liên quan mạnh giữa tiên đề và kết quả thì ta không rút ra được quy luật nào cả.

```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.5)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False, False])
specific_word = 'Age'
filtered_rules = rules[rules['antecedents'].apply(lambda x:  any(specific_word in item for item in x))]
filtered_rules
```

Ta sẽ chỉnh min support của các itemsets xuống 0.01, được các quy luật là:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
1	(Frequency of Purchases=Monthly, Age=18-29)	(Payment Method=Venmo)	0.032461	0.164398	0.010471	0.322581	1.962194	0.005135	1.233508	0.506818
13	(Payment Method=Venmo, Age=40-49)	(Frequency of Purchases=Every 3 Months)	0.035602	0.159162	0.010471	0.294118	1.847910	0.004805	1.191187	0.475787
7	(Frequency of Purchases=Quarterly, Age=18-29)	(Payment Method=Bank Transfer)	0.043979	0.167539	0.012565	0.285714	1.705357	0.005197	1.165445	0.432640
14	(Frequency of Purchases=Every 3 Months, Age=40...	(Payment Method=Venmo)	0.037696	0.164398	0.010471	0.277778	1.689667	0.004274	1.156988	0.424157
8	(Payment Method=Bank Transfer, Age=18-29)	(Frequency of Purchases=Quarterly)	0.047120	0.150785	0.012565	0.266667	1.768519	0.005460	1.158020	0.456044
2	(Payment Method=Venmo, Age=18-29)	(Frequency of Purchases=Monthly)	0.039791	0.139267	0.010471	0.263158	1.889592	0.004930	1.168138	0.490294
17	(Age=40-49)	(Payment Method=Venmo, Frequency of Purchases=...	0.193717	0.031414	0.010471	0.054054	1.720721	0.004386	1.023934	0.519481
11	(Age=18-29)	(Frequency of Purchases=Quarterly, Payment Met...	0.253403	0.031414	0.012565	0.049587	1.578512	0.004605	1.019121	0.490884
5	(Age=18-29)	(Payment Method=Venmo, Frequency of Purchases=...	0.253403	0.024084	0.010471	0.041322	1.715774	0.004368	1.017982	0.558766

Tuy đã rút ra được quy luật nhưng số lượng là rất ít, ta sẽ chỉnh min support của các itemsets xuống 0.009.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
26	(Frequency of Purchases=Monthly, Age=50-59)	(Payment Method=PayPal)	0.023037	0.167539	0.009424	0.409091	2.441761	0.005565	1.408780	0.604383
33	(Frequency of Purchases=Weekly, Age=50-59)	(Payment Method=Bank Transfer)	0.026178	0.167539	0.009424	0.360000	2.148750	0.005038	1.300720	0.548984
15	(Frequency of Purchases=Fortnightly, Age=30-39)	(Payment Method=Cash)	0.027225	0.169634	0.009424	0.346154	2.040598	0.004806	1.269972	0.524220
1	(Frequency of Purchases=Monthly, Age=18-29)	(Payment Method=Venmo)	0.032461	0.164398	0.010471	0.322581	1.962194	0.005135	1.233508	0.506818
44	(Frequency of Purchases=Weekly, Age=60-70)	(Payment Method=Cash)	0.029319	0.169634	0.009424	0.321429	1.894841	0.004451	1.223698	0.486516
16	(Payment Method=Cash, Age=30-39)	(Frequency of Purchases=Fortnightly)	0.030366	0.132984	0.009424	0.310345	2.333695	0.005386	1.257173	0.589393
40	(Age=60-70, Payment Method=Debit Card)	(Frequency of Purchases=Quarterly)	0.030366	0.150785	0.009424	0.310345	2.058190	0.004845	1.231361	0.530238
21	(Payment Method=Venmo, Age=40-49)	(Frequency of Purchases=Every 3 Months)	0.035602	0.159162	0.010471	0.294118	1.847910	0.004805	1.191187	0.475787
38	(Frequency of Purchases=Quarterly, Age=60-70)	(Payment Method=Debit Card)	0.032461	0.165445	0.009424	0.290323	1.754798	0.004054	1.175964	0.444565
7	(Frequency of Purchases=Quarterly, Age=18-29)	(Payment Method=Bank Transfer)	0.043979	0.167539	0.012565	0.285714	1.705357	0.005197	1.165445	0.432640

Diễn giải kết quả quy luật

Chúng ta có thể thấy các khách hàng trong độ tuổi từ 50-59 thường hay sử dụng thẻ ngân hàng hoặc Paypal để mua sắm, chúng ta có thể hợp tác với ngân hàng và Paypal tạo ra các chương trình giảm giá khi thanh toán nhằm vào các đối tượng này để khuyến khích khách hàng chi tiêu nhiều hơn. Các khách hàng trong độ tuổi từ 18-29 chi tiêu thường xuyên hàng tháng thì hay sử dụng Venmo để thanh toán, ta có thể tạo ra các chương trình tích điểm hàng tháng cho phân khúc khách hàng này để khuyến khích họ chi tiêu nhiều hơn mỗi tháng...

3.2 Xây dựng luật kết hợp với giải thuật FP-Growth

3.2.1 Giải thuật FP-Growth

1. **Ý tưởng giải thuật**

- Nén tập dữ liệu vào cấu trúc cây (Frequent Pattern tree, FP-tree)
  - Giảm chi phí cho toàn tập dữ liệu dùng trong quá trình khai phá. Infrequent items bị loại bỏ sớm.
  - Đảm bảo kết quả khai phá không bị ảnh hưởng
- Phương pháp chia-để-trị (divide-and-conquer).
  - Quá trình khai phá được chia thành các công tác nhỏ. Một là xây dựng FP-tree. Hai là khám phá frequent itemsets với FP-tree.
- Tránh tạo ra các tập dự tuyển. Tức mỗi lần kiểm tra một phần tập dữ liệu.

2. Đặc điểm của thuật toán

- Hai giao dịch có chứa cùng một số các mục, thì đường đi của chúng sẽ có phần (đoạn) chung.
- Càng nhiều các đường đi có phần tử chung, thì việc biểu diễn bằng FP-Tree sẽ càng gọn.

3. Mã giả của giải thuật

```
procedure FP_growth(Tree, α)
(1)  if Tree contains a single path P then
(2)      for each combination (denoted as β) of the nodes in the path P
(3)          generate pattern β∪α with support_count = minimum support count of nodes in β;
(4)  else for each a_i in the header of Tree {
(5)      generate pattern β = a_i ∪ α with support_count = a_i.support_count;
(6)      construct β's conditional pattern base and then β's conditional FP_tree Tree_β;
(7)      if Tree_β ≠ ∅ then
(8)          call FP_growth(Tree_β, β); }
```

4. Xây dựng mô hình với FP-Growth

Ta sẽ sử dụng thuật toán FP-Growth để đưa ra các tập phổ biến với min_support = 0.009 (giá trị của min_support giống như với thuật toán Apriori).

```
from mlxtend.frequent_patterns import fpgrowth

frequent_itemsets = fpgrowth(final_df, min_support=0.009, use_colnames=
    True)
frequent_itemsets.tail(30)
```

	support	itemsets
160	0.016340	(Frequency of Purchases=Weekly, Age=25-29)
161	0.016340	(Frequency of Purchases=Quarterly, Age=25-29)
162	0.019608	(Payment Method=Credit Card, Age=25-29)
163	0.013072	(Frequency of Purchases=Every 3 Months, Age=25...
164	0.009804	(Frequency of Purchases=Fortnightly, Age=25-29)
165	0.009804	(Payment Method=Bank Transfer, Age=25-29)
166	0.019608	(Payment Method=Cash, Age=25-29)
167	0.013072	(Frequency of Purchases=Monthly, Age=25-29)
168	0.009804	(Frequency of Purchases=Weekly, Payment Method...
169	0.009804	(Payment Method=Credit Card, Frequency of Purc...
170	0.035948	(Frequency of Purchases=Annually, Age=18-24)
171	0.035948	(Payment Method=Credit Card, Frequency of Purc...
172	0.039216	(Payment Method=PayPal, Frequency of Purchases...
173	0.013072	(Payment Method=Cash, Frequency of Purchases=A...
174	0.013072	(Payment Method=PayPal, Frequency of Purchases...
175	0.042484	(Payment Method=Credit Card, Age=60-64)

Tập phổ biến

176	0.013072	(Frequency of Purchases=Fortnightly, Age=60-64)
177	0.019608	(Age=60-64, Frequency of Purchases=Every 3 Mon...
178	0.013072	(Age=60-64, Payment Method=Venmo)
179	0.022876	(Age=60-64, Frequency of Purchases=Annually)
180	0.013072	(Payment Method=PayPal, Age=60-64)
181	0.013072	(Age=60-64, Frequency of Purchases=Monthly)
182	0.009804	(Age=60-64, Frequency of Purchases=Bi-Weekly)
183	0.013072	(Age=60-64, Payment Method=Debit Card)
184	0.022876	(Frequency of Purchases=Weekly, Age=60-64)
185	0.022876	(Payment Method=Cash, Age=60-64)
186	0.009804	(Payment Method=Credit Card, Age=60-64, Freque...
187	0.009804	(Payment Method=Credit Card, Age=60-64, Freque...
188	0.009804	(Payment Method=Credit Card, Age=60-64, Freque...
189	0.013072	(Payment Method=Cash, Frequency of Purchases=W...

Tập phổ biến

Sau đó ta tiến hành xây dựng luật kết hợp từ tập phổ biến trên. Các bước thực hiện sẽ giống với khi xây dựng luật kết hợp từ thuật toán Apriori.

```
rules = association_rules(frequent_itemsets, metric="lift",
    min_threshold=1)
rules = rules.sort_values(['confidence', 'lift'], ascending =[False,
    False])
```

Tiếp theo ta sẽ lọc ra các luật kết hợp có chứa giá trị tuổi (Age) trong tiên đề.

```
specific_word = 'Age'

filtered_rules = rules[rules['antecedents'].apply(lambda x:  any(
    specific_word in item for item in x))]
```

Và đây chính là tập các luật kết hợp mà chúng ta có.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
150	(Age=55-59, Frequency of Purchases=Quarterly)	(Payment Method=Credit Card)	0.013072	0.202614	0.009804	0.750000	3.701613	0.007155	3.189542	0.739514
254	(Frequency of Purchases=Monthly, Age=25-29)	(Payment Method=Credit Card)	0.013072	0.202614	0.009804	0.750000	3.701613	0.007155	3.189542	0.739514
282	(Frequency of Purchases=Fortnightly, Age=60-64)	(Payment Method=Credit Card)	0.013072	0.202614	0.009804	0.750000	3.701613	0.007155	3.189542	0.739514
107	(Frequency of Purchases=Weekly, Age=35-39)	(Payment Method=PayPal)	0.013072	0.205882	0.009804	0.750000	3.642857	0.007113	3.176471	0.735099
76	(Frequency of Purchases=Weekly, Age=45-49)	(Payment Method=Bank Transfer)	0.019608	0.150327	0.013072	0.666667	4.434783	0.010124	2.549020	0.790000
174	(Frequency of Purchases=Monthly, Age=18-24)	(Payment Method=Cash)	0.019608	0.166667	0.013072	0.666667	4.000000	0.009804	2.500000	0.765000
247	(Frequency of Purchases=Weekly, Age=25-29)	(Payment Method=Debit Card)	0.016340	0.124183	0.009804	0.600000	4.831579	0.007775	2.189542	0.806202
82	(Age=45-49, Frequency of Purchases=Quarterly)	(Payment Method=Bank Transfer)	0.016340	0.150327	0.009804	0.600000	3.991304	0.007348	2.124183	0.761905
216	(Frequency of Purchases=Weekly, Age=30-34)	(Payment Method=Venmo)	0.016340	0.150327	0.009804	0.600000	3.991304	0.007348	2.124183	0.761905
74	(Payment Method=Bank Transfer, Frequency of Pu...	(Age=45-49)	0.022876	0.088235	0.013072	0.571429	6.476190	0.011053	2.127451	0.865385
299	(Payment Method=Cash, Age=60-64)	(Frequency of Purchases=Weekly)	0.022876	0.166667	0.013072	0.571429	3.428571	0.009259	1.944444	0.724916
300	(Frequency of Purchases=Weekly, Age=60-64)	(Payment Method=Cash)	0.022876	0.166667	0.013072	0.571429	3.428571	0.009259	1.944444	0.724916
93	(Payment Method=Credit Card, Age=18-24)	(Frequency of Purchases=Every 3 Months)	0.019608	0.143791	0.009804	0.500000	3.477273	0.006984	1.712418	0.726667
8	(Frequency of Purchases=Fortnightly, Age=18-24)	(Payment Method=Bank Transfer)	0.019608	0.150327	0.009804	0.500000	3.326087	0.006856	1.699346	0.713333
253	(Payment Method=Credit Card, Age=25-29)	(Frequency of Purchases=Monthly)	0.019608	0.153595	0.009804	0.500000	3.255319	0.006792	1.692810	0.706667
24	(Payment Method=PayPal, Age=50-54)	(Frequency of Purchases=Weekly)	0.019608	0.166667	0.009804	0.500000	3.000000	0.006536	1.666667	0.680000
248	(Payment Method=Debit Card, Age=25-29)	(Frequency of Purchases=Weekly)	0.019608	0.166667	0.009804	0.500000	3.000000	0.006536	1.666667	0.680000
288	(Age=60-64, Frequency of Purchases=Every 3 Mon...	(Payment Method=Credit Card)	0.019608	0.202614	0.009804	0.500000	2.467742	0.005831	1.594771	0.606667
192	(Age=40-44, Frequency of Purchases=Monthly)	(Payment Method=PayPal)	0.026144	0.205882	0.013072	0.500000	2.428571	0.007689	1.588235	0.604027

Nhìn chung các luật kết hợp được suy ra sau khi sử dụng thuật toán FP-Growth không khác gì so với các luật kết hợp sử dụng thuật toán Apriori.

3.3 Đánh giá mô hình giữa 2 giải thuật : Apriori và FP-Growth

Sau khi áp dụng các thuật toán Apriori và FP-Growth để tìm kiếm các luật kết hợp, chúng ta có thể rút ra các kết luận về ưu nhược điểm của từng thuật toán như sau:

3.3.1 Ưu Điểm

Thuật toán Apriori

- **Đơn giản và dễ hiểu**: Thuật toán Apriori dễ hiểu và dễ triển khai, phù hợp cho những người mới bắt đầu học về khai phá dữ liệu.
- **Phổ biến rộng rãi**: Là một trong những thuật toán kinh điển, Apriori được sử dụng rộng rãi và hỗ trợ trong nhiều thư viện và công cụ phần mềm.

Thuật toán FP-Growth

- **Hiệu suất cao**: FP-Growth hoạt động hiệu quả hơn Apriori, đặc biệt là với các tập dữ liệu lớn, do không cần quét toàn bộ cơ sở dữ liệu nhiều lần.
- **Tiết kiệm bộ nhớ**: Cấu trúc cây FP giúp tiết kiệm bộ nhớ bằng cách nén các mẫu thường xuyên chung.

3.3.2 Nhược Điểm

Thuật toán Apriori

- **Hiệu suất thấp với dữ liệu lớn**: Do phải quét toàn bộ cơ sở dữ liệu nhiều lần, thuật toán Apriori có thể không hiệu quả với các tập dữ liệu lớn.
- **Tốn kém tài nguyên**: Số lượng các tập hợp mục có thể tăng lên nhanh chóng, làm tăng chi phí tính toán và bộ nhớ.

Thuật toán FP-Growth

- **Phức tạp hơn**: FP-Growth phức tạp hơn trong việc triển khai và yêu cầu hiểu biết về cấu trúc cây FP.
- **Không phù hợp cho dữ liệu phân tán**: FP-Growth có thể gặp khó khăn khi xử lý dữ liệu phân tán hoặc không phù hợp với cấu trúc cây FP.

3.3.3 So Sánh Hai Thuật Toán

Tiêu chí	Đặc điểm	Apriori	FP-Growth
Hiệu quả	Hiệu suất với dữ liệu lớn	Thấp	Cao
Đơn giản	Độ phức tạp khi triển khai	Thấp	Cao
Tiêu tốn tài nguyên	Sử dụng bộ nhớ và CPU	Cao	Thấp
Phổ biến	Mức độ sử dụng rộng rãi	Cao	Trung bình

Bảng 1: So sánh ưu nhược điểm của Apriori và FP-Growth

4 Ứng dụng mô hình vào hiểu người dùng trực tuyến

Hiện nay, với sự phát triển của xã hội, có nhiều nội dung mà người dùng có xu hướng hướng tới một nhóm nội dung hẹp hơn dựa trên sở thích của họ. Vì vậy, chúng ta cần lọc nội dung phong phú dựa trên sở thích của một người dùng. Bằng ý tưởng đó và sử dụng giải thuật Apriori, bọn em thực hiện xây dựng hệ thống đề xuất đơn giản được thực hiện bởi những kỹ thuật đã biết để cung cấp các khuyến nghị chung. Đề xuất những thứ được thịnh hành, xếp hạng cao nhất trong dạnh mục....

4.1 Mô tả ứng dụng với dữ liệu mới

4.1.1 Thông tin bộ dữ liệu mới

Bộ dữ liệu mới mà chúng em sử dụng để ứng dụng thuật toán Apriori có tên là *"Anime Recommendations Database"* có nguồn là Link database.

Thông tin bộ dữ liệu

- Gồm 2 file dữ liệu định dạng csv : *anime*, *rating*.
- Ở đây bọn em sẽ chỉ sử dụng file *anime.csv* với kích thước là 936.46 KB.
- Gồm các trường dữ liệu là : anime_id, name, genre, type, episodes, rating, members.

4.2 Chạy thuật toán Apriori

Thực hiện đọc nguồn dữ liệu

```
df_anime = pd.read_csv('anime.csv')
df_anime
```

	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.3700	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.2600	793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.2500	114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.1700	673572
4	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.1600	151266
...	...	...	...	...	...	...	...
12289	9316	Toushindai My Lover: Minami tai Mecha-Minami	Hentai	OVA	1	4.1500	211
12290	5543	Under World	Hentai	OVA	1	4.2800	183
12291	5621	Violence Gekiga David no Hoshi	Hentai	OVA	4	4.8800	219
12292	6133	Violence Gekiga Shin David no Hoshi: Inma Dens...	Hentai	OVA	1	4.9800	175
12293	26081	Yasuji no Pornorama: Yacchimae!!	Hentai	Movie	1	5.4600	142

12294 rows × 7 columns

Với dữ liệu đầu vào như trên, ta sẽ tiến hành lọc ra các thể loại phim (gene).

```
df_genre = df_anime['genre']
items_by_genre = []
for index, genres in df_genre.items():
    items = frozenset(sorted(str(genres).split(', ')))
    items_by_genre.append(items)

```

```
te = TransactionEncoder()
te_ary = te.fit(items_by_genre).transform(items_by_genre)
df_encoded = pd.DataFrame(te_ary, columns=te.columns_, index=df_anime.
    anime_id)
df_encoded
```

Tiếp theo ta sử dụng thuật toán Apriori để đưa ra các tập phổ biến với min_support= 0.05 (do hiệu chỉnh min_support hạ dần xuống để xuất hiện các tập phổ biến sao cho vừa ý mình). Với cột anime_id sẽ đại diện cho transaction và tất cả các genre phải ở dạng cột.

```
frequent_itemsets = apriori(df_encoded, min_support=0.05, use_colnames=
    True)
frequent_itemsets
```

	support	itemsets
0	0.2314	(Action)
1	0.1910	(Adventure)
2	0.3778	(Comedy)
3	0.1640	(Drama)
4	0.0518	(Ecchi)
5	0.1878	(Fantasy)
6	0.0928	(Hentai)
7	0.0656	(Historical)
8	0.1309	(Kids)
9	0.0633	(Magic)
10	0.0768	(Mecha)
11	0.0700	(Music)
12	0.1191	(Romance)
13	0.0992	(School)
14	0.1684	(Sci-Fi)
15	0.1392	(Shounen)
16	0.0992	(Slice of Life)
17	0.0844	(Supernatural)

Tập phổ biến

18	0.0725	(Adventure, Action)
19	0.0698	(Comedy, Action)
20	0.0573	(Fantasy, Action)
21	0.0841	(Sci-Fi, Action)
22	0.0639	(Shounen, Action)
23	0.0745	(Adventure, Comedy)
24	0.0774	(Adventure, Fantasy)
25	0.0528	(Sci-Fi, Adventure)
26	0.0569	(Adventure, Shounen)
27	0.0707	(Comedy, Fantasy)
28	0.0626	(Comedy, Romance)
29	0.0617	(Comedy, School)
30	0.0515	(Sci-Fi, Comedy)
31	0.0764	(Comedy, Shounen)
32	0.0569	(Comedy, Slice of Life)
33	0.0585	(Sci-Fi, Mecha)

Tập phổ biến

Tiến hàng xây dựng luật kết hợp từ tập phổ biến trên.

```
rules = association_rules(frequent_itemsets, metric='confidence',
    min_threshold=0.3)
rules = rules[rules['lift'] >= 1]
rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(Adventure)	(Action)	0.1910	0.2314	0.0725	0.3795	1.6398	0.0283	1.2386	0.4823
1	(Action)	(Adventure)	0.2314	0.1910	0.0725	0.3132	1.6398	0.0283	1.1779	0.5076
3	(Fantasy)	(Action)	0.1878	0.2314	0.0573	0.3053	1.3194	0.0139	1.1064	0.2981
4	(Sci-Fi)	(Action)	0.1684	0.2314	0.0841	0.4995	2.1585	0.0451	1.5357	0.6454
5	(Action)	(Sci-Fi)	0.2314	0.1684	0.0841	0.3634	2.1585	0.0451	1.3064	0.6983
6	(Shounen)	(Action)	0.1392	0.2314	0.0639	0.4594	1.9851	0.0317	1.4217	0.5765
7	(Adventure)	(Comedy)	0.1910	0.3778	0.0745	0.3901	1.0325	0.0023	1.0202	0.0389
8	(Adventure)	(Fantasy)	0.1910	0.1878	0.0774	0.4050	2.1565	0.0415	1.3651	0.6629
9	(Fantasy)	(Adventure)	0.1878	0.1910	0.0774	0.4119	2.1565	0.0415	1.3756	0.6603
10	(Sci-Fi)	(Adventure)	0.1684	0.1910	0.0528	0.3135	1.6416	0.0206	1.1785	0.4700
11	(Shounen)	(Adventure)	0.1392	0.1910	0.0569	0.4091	2.1421	0.0304	1.3692	0.6194
13	(Romance)	(Comedy)	0.1191	0.3778	0.0626	0.5253	1.3902	0.0176	1.3106	0.3187
14	(School)	(Comedy)	0.0992	0.3778	0.0617	0.6213	1.6444	0.0242	1.6430	0.4351
16	(Shounen)	(Comedy)	0.1392	0.3778	0.0764	0.5488	1.4525	0.0238	1.3789	0.3619
17	(Slice of Life)	(Comedy)	0.0992	0.3778	0.0569	0.5730	1.5164	0.0194	1.4569	0.3781
18	(Sci-Fi)	(Mecha)	0.1684	0.0768	0.0585	0.3473	4.5236	0.0456	1.4145	0.9366
19	(Mecha)	(Sci-Fi)	0.0768	0.1684	0.0585	0.7617	4.5236	0.0456	3.4891	0.8437

4.3 Diễn giải các kết quả

Kết quả chạy ra ta có 19 luật kết hợp, với các chỉ số Lift đều lớn hơn 1. Đây chứng tỏ các thể loại anime có mối quan hệ tích cực.
Để cung cấp đề xuất thể loại anime cho người xem ở giai đoạn "Giỏ hàng" hay mục đề xuất. Chúng ta có thể trực tiếp lấy thông tin có sẵn ngay khi người dùng vào trang web và thực hiện xem thể loại nào đó.

Diễn giải quy luật kết hợp

- Quy tắc *Mecha* -> *Sci-Fi* có mức độ tin cậy cao là 0.7617 và lift rất cao là 4.5236. Điều này cho thấy sự liên kết mạnh mẽ, nghĩa là những người xem thể loại Mecha có khả năng cao sẽ thích thể loại Sci-Fi. Quy tắc *Sci-Fi* -> *Mecha* cũng có mức độ tin cậy cao (0.3473) và lift (4.5236), cho thấy sự liên kết hai chiều mạnh mẽ giữa hai thể loại này.

- Quy tắc *Action* -> *Sci-Fi* và *Sci-Fi* -> *Action* với mức độ tin cậy cao (0.3634 và 0.4995) và lift (2.1585) cho thấy những người quan tâm đến thể loại Action có khả năng cao sẽ thích thể loại Sci-Fi và ngược lại.

- Quy tắc (Sci-Fi) -> (Adventure) có mức độ hỗ trợ cao (0.0528) và lift (1.6416), cho thấy một phần đáng kể người dùng thích thể loại Sci-Fi cũng thích thể loại Adventure. Từ đó ta thấy được thể loại anime phổ biến.

Ứng dụng của các quy luật này

- Sử dụng các quy tắc có mức độ tin cậy cao để gợi ý thể loại. Ví dụ, nếu một người dùng thích "Mecha", hãy gợi ý "Sci-Fi". Sử dụng các quy tắc liên kết mạnh hai chiều (ví dụ: Action <-> Sci-Fi) để gợi ý cả hai thể loại cho những người dùng quan tâm đến một trong hai.

- => Cách hoạt động: Dựa trên các thuộc tính của phim (như thể loại) và sở thích của người dùng đã được biết từ trước (những bộ phim họ đã xem và thích). Ví dụ: Nếu người dùng thích nhiều phim thuộc thể loại khoa học viễn tưởng, hệ thống sẽ gợi ý nhiều phim cùng thể loại này.

5 Kết luận

5.1 Ưu nhược điểm của cách tiếp cận

Ưu điểm:

- Có khả năng tìm ra các quy luật hiếm.
- Giúp con người có thể tìm ra mối quan hệ ẩn giữa các mục dữ liệu, thuộc tính mà ta không thể nhận ra.
- Có rất nhiều ứng dụng trong cuộc sống như : phân tích dữ liệu bán lẻ (market basket analysis), tư vấn trực tuyến (online recommendation), hiểu người dùng trực tuyến (user understanding), phân tích tìm ngoại lệ (outlier detection)....
- Các quy luật kết hợp thường dễ hiểu và có thể diễn đạt một cách đơn giản. Điều này giúp cho việc truyền đạt và ứng dụng các kết quả trở nên thuận tiện hơn.
- Các thuật toán khai phá quy luật kết hợp như Apriori, FP-Growth có thể xử lý các tập dữ liệu lớn và tự động hóa quá trình phát hiện quy luật.

Nhược điểm:

- Với các tập dữ liệu rất lớn, các thuật toán có thể gặp khó khăn về thời gian xử lý và yêu cầu tài nguyên tính toán lớn.
- Xuất hiện nhiều quy luật tầm thường.
- Không phải tất cả quy luật tìm ra đều có ý nghĩa thực tế, hoặc áp dụng được vào thực tế. Nhiều quy luật có sự trùng lặp và không có giá trị sử dụng
- Có thể tạo nhiều luật gây ra quá tải thông tin.
- Do có thể tạo nhiều luật gây ra khó khăn cho việc chọn lọc phân tích các quy luật.
- Các thuật toán khai phá quy luật kết hợp thường yêu cầu người dùng phải cấu hình các thông số như ngưỡng hỗ trợ (support) và độ tin cậy (confidence). Việc lựa chọn các giá trị này có thể ảnh hưởng lớn đến kết quả và yêu cầu kinh nghiệm từ người dùng.
- Không phù hợp cho mọi loại dữ liệu: thường phù hợp với dữ liệu rời rạc và định lượng. Với các loại dữ liệu khác (như dữ liệu liên tục), cần phải thực hiện bước tiền xử lý (như phân loại) để chuyển đổi dữ liệu trước khi áp dụng thuật toán.

5.2 Khả năng ứng dụng của kết quả nghiên cứu trong tương lai

Kết quả nghiên cứu về việc tìm các mối liên hệ giữa các mặt hàng trong cửa hàng bằng thuật toán Apriori và FP-Growth mang lại nhiều tiềm năng ứng dụng trong thực tế:

- **Tối ưu hóa chiến lược kinh doanh:** Các quy tắc liên kết có thể giúp nhà bán lẻ hiểu rõ hơn về hành vi mua hàng của khách hàng. Dựa trên các quy tắc này, họ có thể điều chỉnh vị trí sản phẩm trong cửa hàng, tối ưu hóa việc sắp xếp hàng hóa, từ đó tăng tỷ lệ chuyển đổi và doanh thu.

- **Gợi ý sản phẩm:** Phân tích quy tắc liên kết có thể được áp dụng để cải thiện hệ thống gợi ý sản phẩm trên các nền tảng thương mại điện tử. Bằng cách đề xuất các sản phẩm có liên quan đến sản phẩm mà khách hàng đang xem, họ có thể tăng cơ hội bán thêm.
- **Chiến lược marketing:** Hiểu rõ hành vi mua hàng của khách hàng giúp nhà bán lẻ tối ưu hóa chiến lược marketing. Chẳng hạn, việc phối hợp các chiến dịch quảng cáo và khuyến mãi dựa trên quy tắc liên kết có thể thu hút khách hàng mua thêm các sản phẩm liên quan.
- **Dự đoán nhu cầu sản phẩm:** Phân tích các quy tắc liên kết cũng hỗ trợ trong việc dự đoán xu hướng mua hàng tương lai. Điều này giúp cho việc quản lý tồn kho và dự báo nhu cầu sản phẩm một cách hiệu quả hơn.
- **Nghiên cứu thị trường:** Các mối liên hệ phát hiện được cũng có thể cung cấp thông tin quan trọng về sở thích và xu hướng của khách hàng, từ đó hỗ trợ cho quá trình nghiên cứu thị trường và phát triển sản phẩm mới.

Do đó, kết quả nghiên cứu về tìm association rule không chỉ giúp hiểu sâu hơn về hành vi mua hàng mà còn có tiềm năng ứng dụng rộng rãi trong thực tiễn kinh doanh và nghiên cứu thị trường.

Tài liệu tham khảo

1 https://www.geeksforgeeks.org/apriori-algorithm/

2 https://www.geeksforgeeks.org/ml-frequent-pattern-growth-algorithm/

3 https://www.geeksforgeeks.org/implementing-apriori-algorithm-in-python/