BuoiChieu

December 9, 2025

[1]:
```
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori, association_rules
```

1 Bài 1.

[ ]:
```
file_path = 'GroceryStore-AssociateRules.txt'
dataset = []
with open(file_path, 'r') as f:
    lines = f.readlines()
    for line in lines:
        if line.strip() and '\t' in line and ',' in line: # Lọc dòng hợp lệ
            # Tách bỏ số thứ tự đầu dòng, lấy phần danh sách items
            items = line.strip().split('\t')[1].split(',')
            dataset.append(items)

# Chuyển đổi dữ liệu sang dạng One-Hot (TransactionEncoder)
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)

# chọn ngưỡng Support là 0.2 (20%) và Confidence là 0.6 (60%)
# Lý do: Với 20 giao dịch, Support 0.2 nghĩa là sản phẩm phải xuất hiện ít nhất␣
 ↪4 lần.
MIN_SUPPORT = 0.2
MIN_CONFIDENCE = 0.6

# Tìm các tập phổ biến
frequent_itemsets = apriori(df, min_support=MIN_SUPPORT, use_colnames=True)

# Sinh luật kết hợp
rules = association_rules(frequent_itemsets, metric="confidence",␣
 ↪min_threshold=MIN_CONFIDENCE)

# Sắp xếp theo độ mạnh (Lift) để dễ quan sát
rules = rules.sort_values(by=['lift', 'confidence'], ascending=False)
```

```
# Hiển thị kết quả
print("Tổng số luật tìm được:", len(rules))
print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```

```
Tổng số luật tìm được: 5
    antecedents consequents  support  confidence      lift
4       (MAGGI)       (TEA)      0.2    0.800000  2.285714
2  (CORNFLAKES)    (COFFEE)      0.2    0.666667  1.666667
3       (SUGER)    (COFFEE)      0.2    0.666667  1.666667
0        (MILK)     (BREAD)      0.2    0.800000  1.230769
1       (SUGER)     (BREAD)      0.2    0.666667  1.025641
```

Nhận xét: 1. Cặp đôi tiềm năng nhất (Maggi & Tea): Đây là luật mạnh nhất với **Lift = 2.28**. Khách hàng mua Mì gói (Maggi) có xu hướng rất cao (80%) sẽ mua thêm Trà. Cửa hàng nên xếp hai quầy này cạnh nhau.

2. Nhóm "Bữa sáng" (Coffee): Cà phê thường được mua kèm với **Ngũ cốc (Cornflakes)** hoặc **Đường (Sugar)** với mức độ tương quan khá tốt (Lift ~ 1.67). Đây là combo bữa sáng tiêu chuẩn.

3. Mối quan hệ Sữa & Bánh mì: Mặc dù phổ biến, nhưng mức độ thúc đẩy nhau (Lift = 1.23) chỉ ở mức trung bình, thấp hơn nhiều so với cặp Mì - Trà.

4. Luật yếu (Sugar & Bread): Cặp Đường - Bánh mì có **Lift xấp xỉ 1 (1.02)**. Điều này chứng tỏ hai sản phẩm này gần như độc lập, khách mua cùng nhau chỉ là ngẫu nhiên chứ không phải do sản phẩm này kích thích sản phẩm kia.

2 Bài 2:

[ ]:
```
file_path = 'ThiTNTHPT 2021-TpHCM.csv'
df = pd.read_csv(file_path)

df_khtn = df.dropna(subset=['Lý', 'Hoá', 'Sinh']).copy()

cols_mapping = {
    'Toán': 'T',
    'Ngoại Ngữ': 'AV',
    'Văn': 'V',
    'Lý': 'Ly',
    'Hoá': 'Hoa',
    'Sinh': 'Sinh'
}

# Lọc lấy các cột điểm cần thiết
df_processed = df_khtn[list(cols_mapping.keys())].copy()

# Đổi tên cột theo yêu cầu đề bài (T-AV-V...)
df_processed.rename(columns=cols_mapping, inplace=True)
```

```python
# Đánh số lại SBD để bảo mật (Xóa SBD cũ, tạo index mới)
df_processed.reset_index(drop=True, inplace=True)
df_processed.index.name = 'SBD_Moi' # Đặt tên index là SBD mới
df_processed.reset_index(inplace=True) # Biến index thành cột thực

print("Đã tạo xong tập dữ liệu KHTN với", len(df_processed), "thí sinh.")
print(df_processed.head())

# Quy tắc: Điểm >= 8 thành 1, ngược lại thành 0
# Lưu ý: Chỉ áp dụng lên các cột điểm (từ cột thứ 1 trở đi, bỏ cột SBD_Moi đầu tiên)
score_cols = ['T', 'AV', 'V', 'Ly', 'Hoa', 'Sinh']
df_binary = df_processed.copy()

# Áp dụng logic thay thế
df_binary[score_cols] = (df_binary[score_cols] >= 8).astype(int)

print("\n--- Dữ liệu sau khi chuyển đổi 0/1 ---")
print(df_binary.head())

# KHAI PHÁ LUẬT KẾT HỢP (ASSOCIATION RULES) ---
# 1. Tìm tập phổ biến (Frequent Itemsets)
# min_support = 0.05: Môn/Combo môn đó phải xuất hiện ít nhất ở 5% thí sinh (Vì điểm >=8 khá khó nên để thấp)
frequent_itemsets = apriori(df_binary[score_cols].astype(bool), min_support=0.05, use_colnames=True)

# 2. Sinh luật (Rules)
# min_threshold=0.5: Độ tin cậy ít nhất 50%
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.5)

# Sắp xếp theo Lift giảm dần
rules = rules.sort_values(by=['lift', 'confidence'], ascending=False)

# Hiển thị kết quả
print("\n--- KẾT QUẢ LUẬT KẾT HỢP ---")
print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head(10))
```

```
Đã tạo xong tập dữ liệu KHTN với 50179 thí sinh.
   SBD_Moi    T   AV     V    Ly   Hoa  Sinh
0        0  7.2  8.2  7.50  6.50  7.00  5.25
1        1  8.2  5.0  6.00  5.50  7.25  4.75
2        2  7.6  9.0  5.50  6.25  7.00  5.25
3        3  7.8  9.2  8.00  5.75  6.75  5.25
4        4  7.6  9.0  5.75  7.00  6.50  5.00
```

```
--- Dữ liệu sau khi chuyển đổi 0/1 ---
   SBD_Moi  T  AV  V  Ly  Hoa  Sinh
0        0  0   1  0   0    0     0
1        1  1   0  0   0    0     0
2        2  0   1  0   0    0     0
3        3  0   1  1   0    0     0
4        4  0   1  0   0    0     0

--- KẾT QUẢ LUẬT KẾT HỢP ---
   antecedents consequents   support  confidence      lift
8         (Ly)     (T, AV)  0.078200    0.687577  2.356044
7     (Ly, AV)         (T)  0.078200    0.899587  2.078287
2         (Ly)         (T)  0.099384    0.873839  2.018802
10   (Hoa, AV)         (T)  0.084876    0.816996  1.887479
3        (Hoa)         (T)  0.138066    0.745347  1.721951
6      (T, Ly)        (AV)  0.078200    0.786846  1.630793
4         (Ly)        (AV)  0.086929    0.764325  1.584116
0          (T)        (AV)  0.291835    0.674217  1.397363
1         (AV)         (T)  0.291835    0.604849  1.397363
9     (T, Hoa)        (AV)  0.084876    0.614752  1.274116
```

Sau khi xử lý và chuẩn hóa dữ liệu, tập KHTN còn **50.179 thí sinh**, đảm bảo đủ lớn để khai phá luật kết hợp. Việc chuyển điểm về dạng nhị phân (>=8 → 1, <8 → 0) cho thấy tỷ lệ điểm cao ở các môn tự nhiên không nhiều, nhưng vẫn xuất hiện nhiều mối quan hệ rõ rệt.

Các luật có confidence rất cao cho thấy **thí sinh đạt điểm cao môn Lý thường đồng thời đạt điểm cao ở Toán và Anh Văn**, với luật (Ly → T, AV) có lift lên tới **2.35**, thể hiện mức liên hệ mạnh bất thường. Ngoài ra, tổ hợp (Ly, AV → T) và (Ly → T) đều có confidence trên 87%, cho thấy **năng lực học tốt môn Lý đi kèm với năng lực Toán rất rõ rệt**. Môn Hóa và môn Lý cũng góp phần dự đoán điểm cao môn Toán nhưng mức độ yếu hơn.

Nhìn chung, các luật khai phá được phản ánh mối liên hệ tự nhiên giữa các môn trong tổ hợp KHTN: học sinh giỏi Lý có xu hướng giỏi Toán và Anh Văn, và các môn khoa học tự nhiên hỗ trợ nhau mạnh trong kết quả điểm cao.

Bài 3:

[ ]:
```
df = pd.read_csv('RestaurantDataset.csv', header=None, names=['Area',␣
 ↪'Cuisine', 'Grade'])

# Xử lý chuỗi: Xóa khoảng trắng thừa ở đầu/cuối (trim whitespace)
df = df.apply(lambda x: x.str.strip() if x.dtype == "object" else x)

# Chỉ giữ lại các loại nhà hàng: Chinese, French, American, Italian, Japanese,␣
 ↪Asian
target_cuisines = ['Chinese', 'French', 'American', 'Italian', 'Japanese',␣
 ↪'Asian']
df_filtered = df[df['Cuisine'].isin(target_cuisines)].copy()
```

```python
print(f"Số lượng bản ghi sau khi lọc: {len(df_filtered)}")
print(df_filtered.head())

df_encoded = pd.get_dummies(df_filtered, prefix=['Area', 'Cuisine', 'Grade'])

# 1. Tìm tập phổ biến (Frequent Itemsets)
# Min Support = 0.005 (0.5%): Vì ta muốn tìm các luật ngách
frequent_itemsets = apriori(df_encoded.astype(bool), min_support=0.005,␣
 ↪use_colnames=True)

# 2. Sinh luật (Association Rules)
# Min Confidence = 0.5 (50%): Chỉ lấy các luật có độ tin cậy cao
rules = association_rules(frequent_itemsets, metric="confidence",␣
 ↪min_threshold=0.5)

# Sắp xếp theo Lift (độ mạnh của luật) hoặc Confidence (độ tin cậy)
rules_sorted = rules.sort_values(by=['confidence', 'lift'], ascending=False)

# Chọn các cột quan trọng để hiển thị
display_cols = ['antecedents', 'consequents', 'support', 'confidence', 'lift']
print("\n--- Top 10 Luật Kết Hợp mạnh nhất ---")
print(rules_sorted[display_cols].head(10))

# --- KIỂM TRA CÁC LUẬT CỤ THỂ TRONG VÍ DỤ ---
# Lọc ra các luật liên quan đến 'Japanese' để so sánh với đề bài
print("\n--- Các luật liên quan đến 'Japanese' ---")
japanese_rules = rules_sorted[rules_sorted['antecedents'].apply(lambda x:␣
 ↪'Cuisine_Japanese' in x)]
print(japanese_rules[display_cols].head())
```

```
Số lượng bản ghi sau khi lọc: 83174
        Area   Cuisine Grade
0   BROOKLYN   Chinese     Z
1  MANHATTAN  American     C
3  MANHATTAN  American     A
4  MANHATTAN  American     A
5  MANHATTAN  American     C

--- Top 10 Luật Kết Hợp mạnh nhất ---
                                antecedents         consequents   support  \
28              (Grade_A, Cuisine_French)    (Area_MANHATTAN)  0.011374
4                         (Cuisine_French)    (Area_MANHATTAN)  0.020139
30              (Cuisine_French, Grade_B)    (Area_MANHATTAN)  0.005663
36            (Grade_C, Cuisine_Japanese)    (Area_MANHATTAN)  0.008031
33             (Cuisine_Italian, Grade_C)    (Area_MANHATTAN)  0.010135
42  (Cuisine_American, Area_STATENISLAND)           (Grade_A)  0.011157
```

```
19              (Grade_A, Area_MANHATTAN)  (Cuisine_American)  0.156972
32             (Grade_B, Cuisine_Italian)    (Area_MANHATTAN)  0.019441
8                    (Area_STATENISLAND)           (Grade_A)  0.021004
5                      (Cuisine_Italian)    (Area_MANHATTAN)  0.063517

    confidence      lift
28    0.809932  1.669109
4     0.797999  1.644519
30    0.759677  1.565545
36    0.715970  1.475473
33    0.676565  1.394267
42    0.673929  1.345948
19    0.647009  1.251933
32    0.638373  1.315562
8     0.630914  1.260040
5     0.616597  1.270684

--- Các luật liên quan đến 'Japanese' ---
                        antecedents       consequents   support  confidence  \
36      (Grade_C, Cuisine_Japanese)  (Area_MANHATTAN)  0.008031    0.715970
6                 (Cuisine_Japanese)  (Area_MANHATTAN)  0.041780    0.603194
35      (Grade_B, Cuisine_Japanese)  (Area_MANHATTAN)  0.013430    0.584205
34      (Grade_A, Cuisine_Japanese)  (Area_MANHATTAN)  0.018107    0.573933
40  (Area_QUEENS, Cuisine_Japanese)         (Grade_A)  0.005170    0.517449

        lift
36  1.475473
6   1.243064
35  1.203931
34  1.182763
40  1.033432
```

KẾT LUẬN NGẮN GỌN

Sau khi lọc giữ lại 6 nhóm nhà hàng, dữ liệu còn **83.174 bản ghi**, đủ lớn để khai phá luật kết hợp. Kết quả cho thấy xu hướng phân bố nhà hàng tại New York rất rõ rệt: **các nhà hàng French, Italian và Japanese có khả năng cao tập trung ở khu vực Manhattan**, thể hiện qua nhiều luật có confidence trên 75% và lift > 1.6. Ngược lại, **khu Staten Island có xu hướng xuất hiện nhiều nhà hàng được xếp hạng A**, với luật "Area_STATENISLAND Grade_A" có support cao nhất trong top luật mạnh. Nhìn chung, các luật tìm được phản ánh đúng đặc trưng thực tế: Manhattan là khu vực tập trung nhiều nhà hàng nổi tiếng, trong khi Staten Island có chất lượng đánh giá vệ sinh tốt hơn.