**Lesson 11: Data Mining with Python.**

1. **Lý thuyết**

Sau khi có data, điều đầu tiên và cũng là điều quan trọng nhất đó chính là chúng ta phải khai phá được dữ liệu.

Khai phá ở đây không phải là việc khai phá đơn giản như EDA , mà là những phương pháp khai phá dữ liệu để tìm ra những thông tin ẩn , quy luật ẩn ở trong dữ liệu, mà ở bài học trước , chúng ta đã được học về 2 thuật toán dùng để khai phá dữ liệu , đó là Apriori và Eclat.

Ở tiết học này, chúng ta sẽ được học kỹ hơn về khai phá dữ liệu bằng những thuật toán đó.

Ví dụ : Vào năm 2012, chuỗi siêu thị Target của Mỹ bị một ông bố kiện vì gửi quảng cáo dành cho bà bầu đến con gái của ông, trong khi con gái ông còn đang học cấp 3. Nhưng thật sự là con gái của ông đã có thai nhưng giấu gia đình.

Vậy làm thế nào để Target biết được điều này , đó là nhờ vào việc khai phá dữ liệu và tìm ra được những quy tắc ẩn trong những lần giao dịch của khách hàng. Vì những bà bầu thì sẽ có xu hướng mua món a , món b , món c ,... Nếu một người nữ cứ tiếp tục mua những món như thế thì sẽ có xác suất rất cao rằng họ có bầu.

Đọc thêm tại : [Link](https://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did/?sh=3b8814f66686)

1. **Thuật toán Apriori**

Là thuật toán thực hiện việc tìm các mục hoặc phần tử thường xuyên nhất trong cơ sở dữ liệu và xác định các quy tắc kết hợp giữa các mục, được đề xuất lần đầu tiên vào năm 1994 bởi Rakesh Agrawal và Ramakrishnan Srikant.

Theo thuật toán, bất kỳ tập hợp con của một tập hợp phổ biến thì phổ biến. Tương tự, đối với bất kỳ mục (item) không phổ biến nào, tất cả các tập hợp con của nó cũng phải không phổ biến

Ví dụ : Trong một giỏ hàng , những món nào sẽ có xu hướng bán kèm với nhau ?

Icon

Description automatically generated with medium confidence

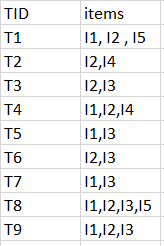
Nếu suy nghĩ logic, thì chúng ta cũng được kết quả là

A picture containing icon

Description automatically generated

Before we start understanding the algorithm, go through some definitions which are explained in my previous post.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

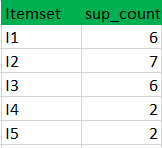


minimum support count is 2

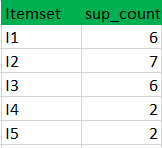
minimum confidence is 60%

**Step-1:** K=1

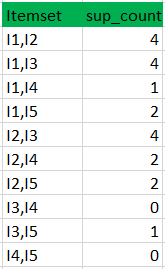
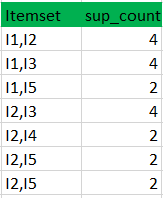
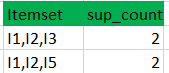
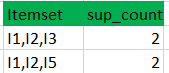
(I) Create a table containing support count of each item present in dataset – Called **C1(candidate set)**

****

(II) compare candidate set item’s support count with minimum support count(here min\_support=2 if support\_count of candidate set items is less than min\_support then remove those items). This gives us itemset L1.



**Step-2:** K=2

* Generate candidate set C2 using L1 (this is called join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common.
* Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)
* Now find support count of these itemsets by searching in dataset.
*   
  (II) compare candidate (C2) support count with minimum support count(here min\_support=2 if support\_count of candidate set item is less than min\_support then remove those items) this gives us itemset L2.  
    
  **Step-3:**
  + Generate candidate set C3 using L2 (join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common. So here, for L2, first element should match.  
    So itemset generated by joining L2 is {I1, I2, I3}{I1, I2, I5}{I1, I3, i5}{I2, I3, I4}{I2, I4, I5}{I2, I3, I5} 5C3
  + Check if all subsets of these itemsets are frequent or not and if not, then remove that itemset.(Here subset of {I1, I2, I3} are {I1, I2},{I2, I3},{I1, I3} which are frequent. For {I2, I3, I4}, subset {I3, I4} is not frequent so remove it. Similarly check for every itemset)
  + find support count of these remaining itemset by searching in dataset.
*   
  (II) Compare candidate (C3) support count with minimum support count(here min\_support=2 if support\_count of candidate set item is less than min\_support then remove those items) this gives us itemset L3.  
    
  **Step-4:**
  + Generate candidate set C4 using L3 (join step). Condition of joining Lk-1 and Lk-1 (K=4) is that, they should have (K-2) elements in common. So here, for L3, first 2 elements (items) should match.
  + Check all subsets of these itemsets are frequent or not (Here itemset formed by joining L3 is {I1, I2, I3, I5} so its subset contains {I1, I3, I5}, which is not frequent). So no itemset in C4
  + We stop here because no frequent itemsets are found further

Nhưng làm thế nào để ta dạy cho máy móc cũng làm giống như vậy ?

À , thì chúng ta sẽ cho nó đọc dữ liệu mua hàng , sau đó nó sẽ thay ta tìm ra quy luật giữa các item

Table

Description automatically generated

Máy móc sẽ đánh giá dựa vào 4 chỉ số chính :

**Support :** là tần số xuất hiện của item / tổng số transaction.

Text

Description automatically generated

**Confidence :** là xác suất của 1 item được mua khi item kia cũng được mua

Chart

Description automatically generated

**Lift :** là correlation của các item.

Lift( x theo y ) = 1 , thì 2 item không có tương quan

Lift ( x theo y ) > 1 , thì 2 item có sự tương quan

Lift ( x theo y ) < 1 , thì 2 item có sự tương quan nghịch, cái này mua thì cái kia k đc mua.

Chart

Description automatically generated

Ví dụ : Khi mua rượu vang và bánh mì rồi thì có mua bánh mì không thì là 2 item không có sự tương quan nên không thể kết luận được.

Conviction

Conviction là quy luật chúng ta cần tìm

Graphical user interface

Description automatically generated with low confidence

Conv dao động từ [0, +∞]

Conv(x theo y ) = 1 thì không có quy luật nào giữa x và y

Conv càng cao thì quy luật càng mạnh

Practice : [Link](https://colab.research.google.com/drive/1mO4rwYAskfZA8R0yYoNzjWilFhILerYQ?usp=sharing)

Apriori có nhược điểm lớn về hiệu suất, đặc biệt khi phải thực hiện thuật toán với những giao dịch phổ biến chưa nhiều đồ vật, như vậy, việc đọc dữ liệu các giao dịch D sẽ phải diễn ra nhiều lần, tốn thời gian và tài nguyên.

**Một vài phương pháp khắc phục:**

Kỹ thuật phân nhóm (hashed-base): chia nhỏ tập các đồ vật thành các nhóm có thể được sử dụng để làm giảm số lượng các giao dịch chứa k đồ vật

Thu nhỏ số lượng giao dịch (transaction reduction) : với giao dịch có k đồ vật và không phải là giao dịch phổ biến thì k đồ vật này sẽ không thể nằm trong k+1 đồ vật của một giao dịch phổ biến được. Do đó, các giao dịch không phổ biến này có thể được loại bỏ khỏi dữ liệu giao dịch D, trong các lần kiểm tra giao dịch chưa nhiều hơn k đồ vật tiêp theo.

Tạo vách ngăn (partitioning): phương pháp này được sử dụng để chia dữ liệu giao dịch D thành các vách nhỏ (n partitions) không trùng lặp nhau

Phương pháp chọn tập con (sample): ý tưởng của việc chọn tập con (S) của toàn bộ dữ liệu giao dịch D là để sử dụng S làm đại diện cho D, sau đó, việc tim giao dịch phổ biến chỉ cần diễn ra trên S sau đó áp dụng ra cho D

1. **Eclat (Equivalence Class Transformation)**

Đối với phương pháp Apriori, thì dữ liệu được sử dụng dưới định dạng nằm ngang, có nghĩa là danh sách các đồ vật trong giao dịch được liệt kê theo thứ tự từ trái sang phải trên cùng một dòng

Cách tiếp cận theo chiều dọc này của thuật toán ECLAT làm cho nó trở thành một thuật toán nhanh hơn so với thuật toán Apriori.

Ví dụ : Lưu trữ giao dịch trong Apriori và Eclat

**Apriori**

Table

Description automatically generated

**Eclat**

Table

Description automatically generated

Eclat và Apriori đều xem xét yếu tố Support và Confidence để tìm ra quy luật

Tuy nhiên, vì cách sắp xếp dữ liệu nên Eclat không cần phải quét dữ liệu toàn bộ dữ liệu cho mỗi lần tính toán như Apriori nên Eclat chạy nhanh hơn và đơn giản hơn.

more: <https://www.geeksforgeeks.org/ml-eclat-algorithm/>