CNTT

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**BÁO CÁO BÀI TẬP LỚN**

**ĐỀ TÀI CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

**NĂM 2020**

**GIẢI THUẬT APRIORI**

**(KHAI PHÁ LUẬT KẾT HỢP)**

**Sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Nguyễn Trọng Thắng | CNTT3-K58 | CNTT |
| Chu Trần Đại | CNTT3-K58 | CNTT |
| Nguyễn Hữu Hưng | CNTT3-K58 | CNTT |
| Trần Thành Long | CNTT3-K58 | CNTT |
| Phạm Quốc Thịnh | CNTT3-K58 | CNTT |

**Người hướng dẫn:** T.S Nguyễn Quốc Tuấn

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**BÁO CÁO TỔNG KẾT**

**ĐỀ TÀI CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

**NĂM 2020**

**GIẢI THUẬT APRIORI**

**(KHAI PHÁ LUẬT KẾT HỢP)**

**Sinh viên thực hiện**

|  |  |  |
| --- | --- | --- |
| Nguyễn Trọng Thắng | Nam, Nữ: Nam | Dân tộc: Kinh |
| Lớp: CNTT3 | Khoa: CNTT | Năm thứ: 4/4 |
| Ngành học: CNTT |  |  |
| Chu Trần Đại | Nam, Nữ: Nam | Dân tộc: Kinh |
| Lớp: CNTT3 | Khoa: CNTT | Năm thứ: 4/4 |
| Ngành học: CNTT |  |  |
| Nguyễn Hữu Hưng | Nam, Nữ: Nam | Dân tộc: Kinh |
| Lớp: CNTT3 | Khoa: CNTT | Năm thứ: 4/4 |
| Ngành học: CNTT |  |  |
| Trần Thành Long | Nam, Nữ: Nam | Dân tộc: Kinh |
| Lớp: CNTT3 | Khoa: CNTT | Năm thứ: 4/4 |
| Ngành học: CNTT |  |  |
| Phạm Quốc Thịnh | Nam, Nữ: Nam | Dân tộc: Kinh |
| Lớp: CNTT3 | Khoa: CNTT | Năm thứ: 4/4 |
| Ngành học: CNTT |  |  |

**Người hướng dẫn:** T.S Nguyễn Quốc Tuấn

Tóm Tắt

Ngày nay, công nghệ thông tin đang phát triển mạnh mẽ và đi vào từng ngóc ngách của đời sống từng ngành nghề. Đi cùng với việc này là việc bùng nổ dữ liệu (big data) là một vấn đề hiện nay. Đây có thể được coi là vùng đất mới cho các nhà phát triển công nghệ. Tương lai không xa nữa thì việc khai thác dữ liệu sẽ là điều không thể thiếu trong công việc của từng nhà phát triển. Vì vậy việc khai thác những tinh hoa từ bộ dữ liệu khổng lồ sẽ là miếng bánh mà được các nhà phát triển nhắm tới. Dữ liệu có thể được khai thác theo nhiều cách khác nhau, theo nhiều mục đích khác nhau để phục vụ cho nhiều vấn đề thiết yếu của xã hội, người dùng. Tại đây, chúng tôi giới thiệu một thuật toán đơn giản và dễ hiểu cấp thiết cho đời sống hiện nay là giải thuật apriori (khai phá luật kết hợp).

**Mục lục**

**CHƯƠNG 1: TỔNG QUAN**

**1.1. Tên đề tài……………………………………………………………………5**

**1.2. Mục tiêu đề tài………………………………………………………………5**

**CHUONG 2: CƠ SỞ LÍ THUYẾT VÀ CÔNG CỤ SỬ DỤNG**

**2.1. Cơ sở lí thuyết………………………………………………………………5**

**2.1.1. Mô tả lý thuyết về luật kết hợp ……………………………………….5**

**2.1.2. Mô tả lý thuyết về giải thuật Apriori………………………………….6**

**2.1.3. Ví dụ trên thuật toán ………………………………………………….6**

**2.2. Công cụ sử dụng**

**2.2.1. Spyder …………...……………………………………………………..9**

**2.2.2. Thư viện mlxtend ………………………………………….…………..9**

**CHƯƠNG 3: Ứng dụng**

**3.1. Tên ứng dụng...……………………………………………………………11**

**3.2. Nguồn dữ liệu ……………………………………………………………..12**

**3.3. Cách thức thực hiện ……………………,………………………………..12**

**3.4. Kết quả thu được……..……………………………….…………………..12**

**CHƯƠNG 4: Mở rộng và kết luận……………………………………………..20**

**CHƯƠNG 5: Tài liệu tham khảo ..……………………………………………..20**

**CHƯƠNG 1: TỔNG QUAN**

**1.1. Tên đề tài:**

Giải thuật Apriori (khai phá luật kết hợp)

**1.2. Mục tiêu đề tài**

Tìm hiểu tổng quan về luật kết hợp. Tìm hiểu giải thuật Apriori. Phân tích, khai phá và ứng dụng dữ liệu bằng cách thức sử dụng giải thuật Apriori.

**CHƯƠNG 2: CƠ SỞ LÍ THUYẾT VÀ CÔNG CỤ SỬ DỤNG**

**2.1.**  **Mô tả lý thuyết về luật kết hợp**

**2.1.1. Mô tả lý thuyết về luật kết hợp**

**Luật kết hợp**

Cho I={I1, I2, .., Im} là tập hợp của m tính chất riêng biệt. Giả sử D là CSDL, với các bản ghi chứa một tập con T các tính chất (có thể coi như Τ ⊆ Ι), các bản ghi đều có chỉ số riêng. Một luật kết hợp là một mệnh đề kéo theo có dạng X→Y, trong đó X, Y ⊆ I, thỏa mãn điều kiện X∩Y=∅. Các tập hợp X và Y được gọi là các tập hợp tính chất (itemset). Tập X gọi là nguyên nhân, tập Y gọi là hệ quả.

Có 2 độ đo quan trọng đối với luật kết hợp: Độ hỗ trợ (support) và độ tin cậy (confidence), được định nghĩa như phần dưới đây.

**Độ hỗ trợ**

Độ hỗ trợ của một tập hợp X trong cơ sở dữ liệu D là tỷ số giữa các bản ghi T ⊆D có chứa tập X và tổng số bản ghi trong D (hay là phần trăm của các bản ghi trong D có chứa tập hợp X), ký hiệu là support(X) hay supp(X) (support sẽ tự sinh ra khi cài thuật toán).



Ta có: 0 ≤supp(X) ≤1 với mọi tập hợp X.

**Hoặc**

Độ hỗ trợ của một luật kết hợp X→Y là tỷ lệ giữa số lượng các bản ghi chứa tập hợp X ∪ Y, so với tổng số các bản ghi trong D - Ký hiệu supp(X→Y).

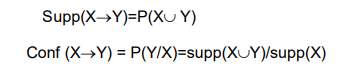


Khi chúng ta nói rằng độ hỗ trợ của một luật là 50%, có nghĩa là có 50% tổng số bản ghi chứa X ∪Y. Như vậy, độ hỗ trợ mang ý nghĩa thống kê của luật.

**Độ tin cậy**

Độ tin cậy của một luật kết hợp X→Y là tỷ lệ giữa số lượng các bản ghi trong D chứa X ∪Y với số bản ghi trong D có chứa tập hợp X. Ký hiệu độ tin cậy của một luật là conf(r). Ta có 0 ≤ conf(r) ≤1

Nhận xét: Độ hỗ trợ và độ tin cậy có xác suất sau:



**Hoặc**

Độ tin cậy của một luật kết hợp X→Y là tỷ lệ giữa số lượng các bản ghi của tập hợp chứa X ∪ Y, so với tổng số các bản ghi chứa X.

Chúng ta nhận thấy rằng tri thức đem lại bởi luật kết hợp dạng trên có sự khác biệt rất nhiều so với những thông tin thu được từ các câu lệnh truy vấn dữ liệu thông thường như SQL. Đó là những tri thức, những mối liên hệ chưa biết trước và mang tính dự báo đang tiềm ẩn trong dữ liệu. Những tri thức này không đơn giản là kết quả của phép nhóm, tính tổng hay sắp xếp mà là của một quá trình tính toán khá phức tạp.

**Tập hợp thường xuyên**

Tập hợp X được gọi là tập hợp thường xuyên (Frenquent itemset) nếu có supp(X) ≥minsup, với minsup là ngưỡng độ hỗ trợ cho trước. Kí hiệu các tập này là FI

**Tính chất 1:** Giả sử A,B ⊆ I là hai tập hợp với A⊆B thì supp(A) ≥supp(B). Như vậy, những bản ghi nào chứa tập hợp B thì cũng chứa tập hợp A

**Tính chất 2:** Giả sử A, B là hai tập hợp, A,B ⊆ I, nếu B là tập hợp thường xuyên và A⊆B thì A cũng là tập hợp thường xuyên. Thật vậy, nếu B là tập hợp thường xuyên thì supp(B) ≥minsup, mọi tập hợp A là con của tập hợp B đều là tập hợp thường xuyên trong cơ sở dữ liệu D vì supp(A) ≥supp(B) (Tính chất 3.1)

**Tính chất 3:** Giả sử A, B là hai tập hợp, A ⊆ B và A là tập hợp không thường xuyên thì B cũng là tập hợp không thường xuyên

Một tập mục X được gọi là đóng (closed) nếu không có tập cha nào của X có cùng độ hỗ trợ với nó, tức là không tồn tại một tập mục X’ nào mà X’⊃X và t(X) = t(X’) (với t(X) và t(X’) tương ứng là tập các giao chứa tập mục X và X’). Ký hiệu tập phổ biến đóng là FCI.

Nếu X là phổ biến và không tập cha nào của X là phổ biến, ta nói rằng X là một tập phổ biến lớn nhất (maximally frequent itemset). Ký hiệu tập tất cả các tập phổ biến lớn nhất là MFI. Dễ thấy MFI ⊆ FCI ⊆ FI.

Khai phá luật kết hợp là công việc phát hiện ra (tìm ra, khám phá, phát hiện) các luật kết hợp thỏa mãn các ngưỡng độ hỗ trợ (δ) và ngưỡng độ tin cậy (α) cho trước. Bài toán khai phá luật kết hợp được chia thành hai bài toán nhỏ, hay như người ta thường nói, việc giải bài toán trải qua hai pha:

* Pha 1: Tìm tất cả các tập phổ biến (tìm FI) trong CSDL T.
* Pha 2: Sử dụng tập FI tìm được ở pha 1 để sinh ra các luật tin cậy (interesting rules).

Ý tưởng chung là nếu gọi ABCD và AB là các tập mục phổ biến, thì chúng ta có thể xác định luật AB →CD với tỷ lệ độ tin cậy:



Nếu conf ≥ minconf thì luật được giữ lại (và thỏa mãn độ hỗ trợ tối thiểu vì ABCD là phổ biến).

Khi các mẫu phổ biến (frequent patterm) dài có từ 15 đến 20 items) thì tập FI, thậm chí cả tập FCI trở nên rất lớn và hầu hết các phương pháp truyền thống phải đếm quá nhiều tập mục mới có thể thực hiện được. Các thuật toán dựa trên thuật toán Apriori – đếm tất cả 2 k tập con của mỗi k- itemsets mà chúng quét qua, và do đó không thích hợp với các itemsets dài được. Các phương pháp khác sử dụng “lookaheads” để giảm số lượng tập mục được đếm. Tuy nhiên, hầu hết các thuật toán này đều sử dụng tìm kiếm theo chiều rộng. Cách làm này hạn chế hiệu quả của lookaheads, vì các mẫu phổ biến dài hơn mà hữu ích vẫn chưa được tìm ra.

**2.1.2. Mô tả lý thuyết về giải thuật Apriori**

**Giải thuật Apriori là gì?**

Thuật toán Apriori được công bố bởi R. Agrawal và R. Srikant vào năm 1994 vì để tìm các tập phổ biến trong một bộ dữ liệu lớn. Tên của thuật toán là Apriori vì nó sử dụng kiến thức đã có từ trước (prior) về các thuộc tính, đối tượng thường xuyên xuất hiện trong cơ sở dữ liệu. Để cải thiện hiệu quả của việc lọc các mục thường xuyên theo cấp độ, một thuộc tính quan trọng được sử dụng gọi là thuộc tính Apriori giúp giảm phạm vi tìm kiếm của thuật toán.

**Thuộc tính Apriori**

Tất cả các tập hợp con không rỗng của tập thường xuyên cũng phải thường xuyên. Khái niệm chủ chốt này của thuật toán Apriori nhằm chống lại đơn điệu của phương pháp tính theo độ hỗ trơ (support). Apriori cho rằng: Tất cả các tập con của một tập hợp thường xuyên phải là thường xuyên (thuộc tính Apriori).Trong một vật phẩm không thường xuyên, tất cả các tập cha của nó sẽ không thường xuyên. Hãy xem xét các tập dữ liệu sau đây và chúng ta sẽ tìm thấy các tập thường xuyên và tạo quy tắc kết hợp cho chúng.

**Ứng dụng của Apriori**

Giải thuật Apriori đơn giản nhưng cực kỳ hữu ích trong nhiều lĩnh vực. Việc thuật toán Apriori có thể làm là nhìn vào quá khứ và khẳng định rằng nếu một việc gì đó xảy ra thì sẽ có tỉ lệ bao nhiêu phần trăm sự việc tiếp theo sẽ xảy ra. Nó giống như nhìn vào quá khứ để dự đoán tương lại vậy, và việc này rất có ích cho các nhà kinh doanh. Ví dụ một siêu thị muốn nghĩ cách sắp xếp các gian hàng một cách hợp lí nhất, họ có thể nhìn vào lịch sử mua hàng và sắp sếp các tập sản phẩm thường được mua cùng nhau vào một chỗ. Hoặc một trang web tin tức muốn giới thiệu cho người dùng các bài viết liên quan đến nhau nhất, cũng có thể áp dụng quy luật tương tự.

Apriori rất tiện dụng, nhưng nó cũng có khá nhiều khuyết điểm:

* Phải duyệt CSDL nhiều lần. Với I = {i\_1, i\_2, ..., i\_{100}}, số lần duyệt CSDL sẽ là 100.
* Số lượng tập ứng viên rất lớn: 2^{100} - 1 = 1.27 \* 10^{30}.
* Thực hiện việc tính độ phổ biến nhiều, đơn điệu.

Cải tiến Apriori : ý tưởng chung

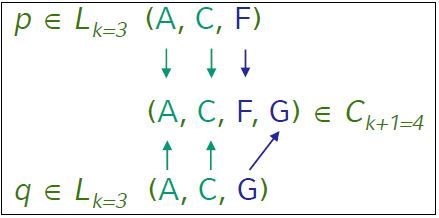
* Giảm số lần duyệt CSDL
* Giảm số lượng tập ứng viên
* Quy trình tính độ phổ biến thuận tiện hơn

1. **Thực hiện và phát triển giải thuật gồm 3 bước để khai thác được các tập phổ biến**

**Bước khởi tạo và kiểm tra**: Khởi tạo tập các ứng cử viên 1th-itemset C1 là tất cả các item có trong bài toán. Tìm các tập phổ biến 1-itemset L1 bằng cách quét cơ sở dữ liệu và xóa tất cả các phần tử không thỏa mãn độ hỗ trợ tối thiểu minSup khỏi C1.

**Bước kết hợp - Join Step:** Tạo ra các tập ứng cử viên candidate itemset ở cấp độ tiếp theo:

* Giả sử ta có 2 tập phổ biến trước đó là k-itemset p và q.
* p và q có thể kết hợp với nhau để tạo ra tập ứng cử viên k+1-itemset nếu chúng có chung k-1 item đầu tiên.

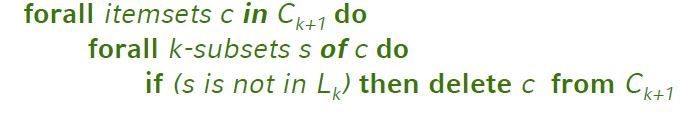


**Bước cắt tỉa – Pruning:**

*Ý tưởng* : Kiểm tra hết tất cả độ phổ biến của mọi itemset ứng cử viên Ck+1 => Không hiệu quả với số lượng lớn các tập Ck+1 được sinh ra.

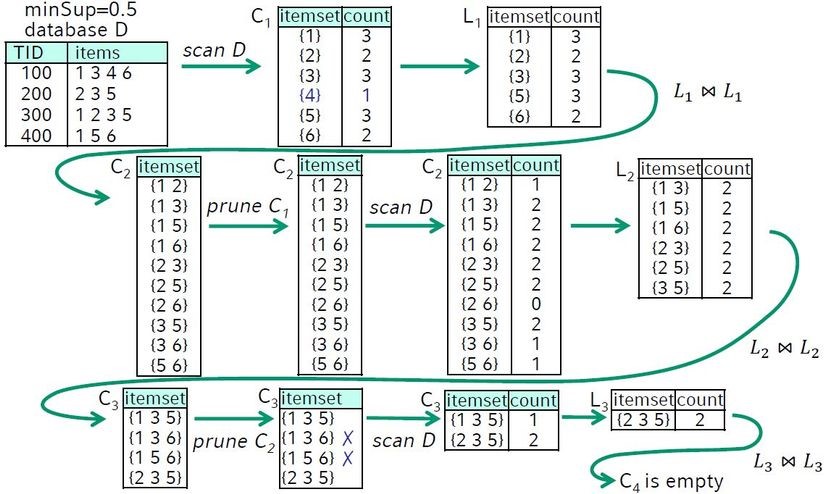
*Giải pháp thay thế*: Áp dụng nguyên tắc thứ nhất của giải thuật Apriori đã nêu ở trên: Loại bỏ tất cả các tập ứng cử viên (k+1) - itemsets mà chứa các tập không phổ biến k-subset:

* + L3 = {(ACF), (ACG), (AFG), (AFH), (CFG)}
  + Tập ứng cử viên sau bước kết hợp: {(ACFG), (AFGH)}
  + Trong bước cắt xén: Xóa tập (AFGH) do có tập con là (FGH) không thuộc L3 ( tập phổ biến 3-itemset sinh ra ở bước trước đó); Tương tự với tập (AGH) không thuộc L3.
  + C4 = {(ACFG)} có các tập con đều thỏa mãn nguyên tắc thứ nhất => Kiểm tra độ phổ biến để sinh ra L4.



Bước 2 và 3 sẽ được lặp đi lặp lại cho đến khi không thể sinh thêm các tập ứng viên được nữa.

1. **Ví dụ đầy đủ áp dụng thuật toán Apriori**

****

Một chú ý nhỏ cho các bạn là các thuật toán trên chỉ áp dụng cho các cơ sở dữ liệu kiểu giao dịch. Tức là các item có hay không có xuất hiện trong một giao dịch (dạng 0,1). Do đó với các item có nhiều giá trị thì cần sử dụng các kĩ thuật nhờ rời rạc hóa, phân cụm, hay áp dụng các kĩ thuật trong logic mờ.... để có thể áp dụng được thuật toán khai thác tập phổ biến.

**2.1.3. Ví dụ trên bài toán**

Đề bài : A database has five transactions. Let min sup = 60% and min conf = 80%

|  |  |
| --- | --- |
| TID | items bought |
| T100 | {M, O, N, K, E, Y} |
| T200 | {D, O, N, K, E, Y } |
| T300 | {M, A, K, E} |
| T400 | {M, U, C, K, Y} |
| T500 | {C, O, O, K, I, E} |

Find all frequent itemsets using Apriori.

**Bước 1:** K = 1 *(I)* Tạo bảng chứa số support của từng mục có trong tập dữ liệu - Được gọi là C1 (tập ứng cử viên)

|  |  |
| --- | --- |
| Itemset | Sup\_count |
| M | 3 |
| O | 4 |
| N | 2 |
| K | 5 |
| E | 4 |
| Y | 3 |
| D | 1 |
| A | 1 |
| U | 1 |
| C | 2 |
| I | 1 |

*(II)* so sánh số support của tập các ứng cử viên với số lượng hỗ trợ tối thiểu (ở đây min\_support = 60% nếu supp(x)(supp(x)=sup\_count/D. X là các itemset trong bảng) của tập ứng cử viên nhỏ hơn min\_support sẽ xóa các tập đó). Điều này cung cấp cho chúng ta mục L1.

|  |  |
| --- | --- |
| Itemset | Sup\_count |
| M | 3 |
| O | 4 |
| K | 5 |
| E | 4 |
| Y | 3 |

**Bước 2:** K = 2 Tạo tập ứng viên C2 bằng L1 (đây được gọi là bước kết hợp). Điều kiện để có thể kết hợp Lk-1 với Lk-1 là hai tập cha đó phải có K-2 (trong trường hợp này là 0) yếu tố chung . Duyệt qua các tập cha của C2, nếu tập cha nào không đạt chuẩn thường xuyên thì tập con đó sẽ bị xóa

|  |  |
| --- | --- |
| Itemset | Sup\_count |
| {M,O} | 1 |
| {M,K} | 3 |
| {M,E} | 2 |
| {M,Y} | 2 |
| {O,K} | 4 |
| {O,Y} | 2 |
| {O,E} | 4 |
| {K,E} | 4 |
| {K,Y} | 3 |
| {E,Y} | 2 |

Tiếp tục kiểm tra độ thường xuyên của các tập trong C2, nếu tập nào không thỏa mãn min\_support thì xóa đi. Ta sẽ nhận được kết quả là tập L2.

|  |  |
| --- | --- |
| Itemset | Sup\_count |
| {M,K} | 3 |
| {O,K} | 4 |
| {O,E} | 4 |
| {K,E} | 4 |
| {K,Y} | 3 |

**Bước 3:** K = 3

* + - Lặp lại quy trình như bước 2 ta được hai tập.
    - C3

|  |  |
| --- | --- |
| Itemset | Sup\_count |
| {M,O,K} | 1 |
| {M,O,E} | 1 |
| {M,O,Y} | 1 |
| {M,K,E} | 2 |
| {M,K,Y} | 2 |
| {M,E,Y} | 1 |
| {O,K,E} | 4 |
| {O,K,Y} | 2 |
| {O,E,Y} | 2 |
| {K,E,Y} | 2 |

* L3

|  |  |
| --- | --- |
| Itemset | Sup\_count |
| {O,K,E} | 4 |

Thực hiện các lần lặp tiếp theo ta thấy không còn tập Cn nào thỏa mãn => L3 là kết quả của lần lặp cuối cùng.

Vậy cuối cùng các hạng mục thường xuyên suất hiện cùng nhau là O,K,E

Bài áp mô phòng đơn giản có thể áp dụng lên bộ dữ liệu mua hàng hóa

Ví dụ O là sữa K là bánh mì E là bơ thì với sự tự tin - độ tin cậy min conf = 80% có nghĩa là 80% khách hàng mà mua sữa và bánh mì cũng sẽ mua bơ.

**2.2. Công cụ sử dụng**

**2.2.1: Spyder – Anaconda**

Spyder là một môi trường khoa học mã nguồn mở miễn phí được viết bằng Python, dành cho Python, được thiết kế bởi và dành cho các nhà khoa học, kỹ sư và nhà phân tích dữ liệu. Nó có sự kết hợp độc đáo giữa chức năng chỉnh sửa, phân tích, gỡ lỗi và lập hồ sơ nâng cao của một công cụ phát triển toàn diện với khả năng khám phá dữ liệu, thực thi tương tác, kiểm tra sâu và khả năng trực quan hóa của một gói khoa học. Nó được tích hợp cùng với nhiều IDE khác trên bộ cài Anaconda.

**2.2.2: Thư viện mlxtend**

Đề cài đặt thư viện mlxtend cùng chạy lệnh **pip install mlxtend. Đây là thư viện mà trong bài này tôi sử dụng để xây dựng mô hình và training data cho giải thuật apriori.**

**Chương 3: Ứng dụng**

**3.1. Tên ứng dụng**

Phân tích, xử lý, thống kê dữ liệu từ các đơn hàng bán lẻ trực tuyến.

**3.2. Nguồn dữ liệu**

Tập dữ liệu bán lẻ trực tuyến.

Đây là tập dữ liệu xuyên quốc gia chứa tất cả các giao dịch xảy ra từ ngày 01/12/2010 đến 09/12/2011 cho một cửa hàng bán lẻ trực tuyến không có cửa hàng và có trụ sở tại Vương quốc Anh.

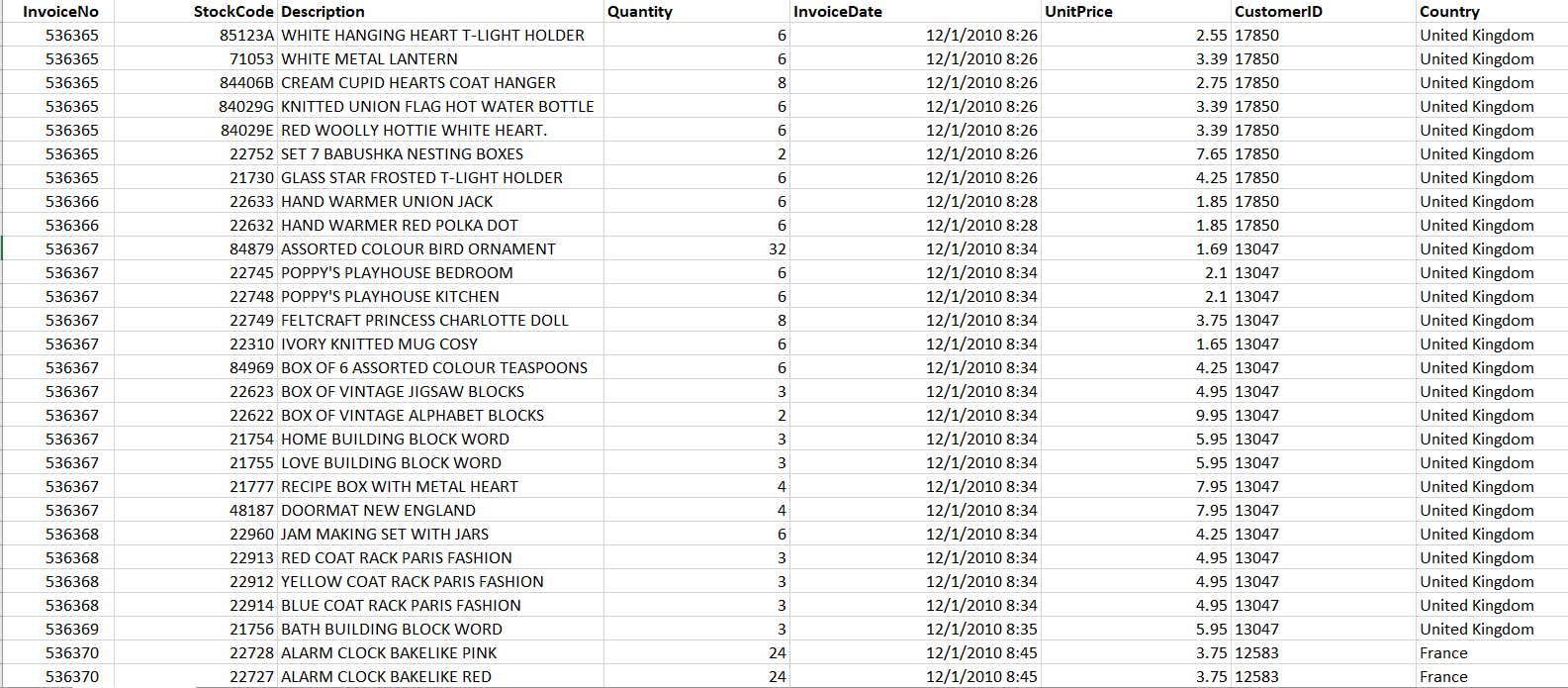
Chi tiết xem tại http://archive.ics.uci.edu/ml/datasets/Online+Retail

**3.3. Cách thức thực hiện**

Để xử bộ dữ liệu và đưa ra những thông số đáng tin cậy về bộ dữ liệu trên thì tôi phát triển trên ngôn ngữ python. sử dụng thư viện mlxtend.

Đây là bộ dữ liệu thực tế sau khi thu thập được. Bộ dữ liệu gồm có 541910 records. Với bộ dữ liệu khá lớn như này thì chúng tôi khá yên tâm trong quá trình xử lý vào training bộ dữ liệu này.

Bên dưới đây là hình ảnh về dữ liệu và các thuộc tính của bộ dữ liệu mà chúng tôi thu thập được và để dùng để xây dựng mô hình giải thuật Apriori và training giải thuật này.

****

**Bước 1:** Chúng tôi sẽ tải dữ liệu vào môi trường để xử lý. Với việc dữ liệu được lưu dưới dạng file excelthì chúng tôi sử dụng thư viện pandas để tải dữ liệu vào môi trường, đây là thư viện thường được các nhà phát triển sử dụng để tải và lấy dữ liệu.

**Bước 2:** Sau khi dữ liệu đã được tải vào môi trường phát triển. Tại bước này thì chúng tôi sẽ thực hiện bước tiền xử lý dữ liệu gồm các công đoạn cụ thể như sau:

* + Làm sạch dữ liệu bằng cách loại bỏ các dữ liệu thiếu thông tin, loại bỏ các thuộc tính không giúp ích cho quá trình khai phá và phân tích dữ liệu.
  + Phân tách dữ liệu còn được gọi là phân nhóm.
  + Mã hóa, biến đổi dữ liệu.

**Bước 3:** Tại bước này thì chúng tôi sẽ đưa dữ liệu vào để xây dựng mô hình apriori và training mô hình sử dụng tập dữ liệu vừa được tiền xử lý ở bước trên.

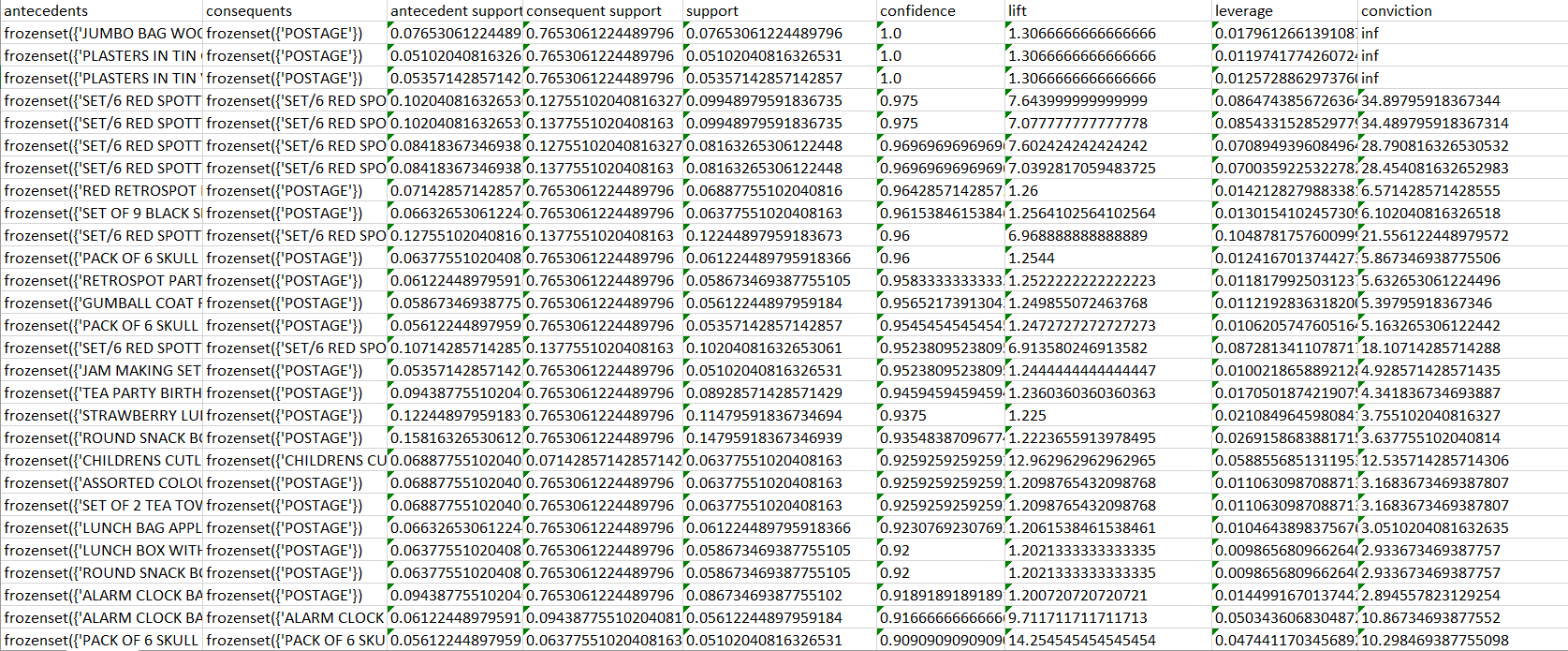
**Bước 4:** Bước này thì chúng tôi sẽ thu được kết quả và đưa ra những nhận định bằng việc phân tích và phân cụm dữ liệu ở trên. Kết quả thu được sẽ được đúc kết tại phần 3.4 tiếp theo.

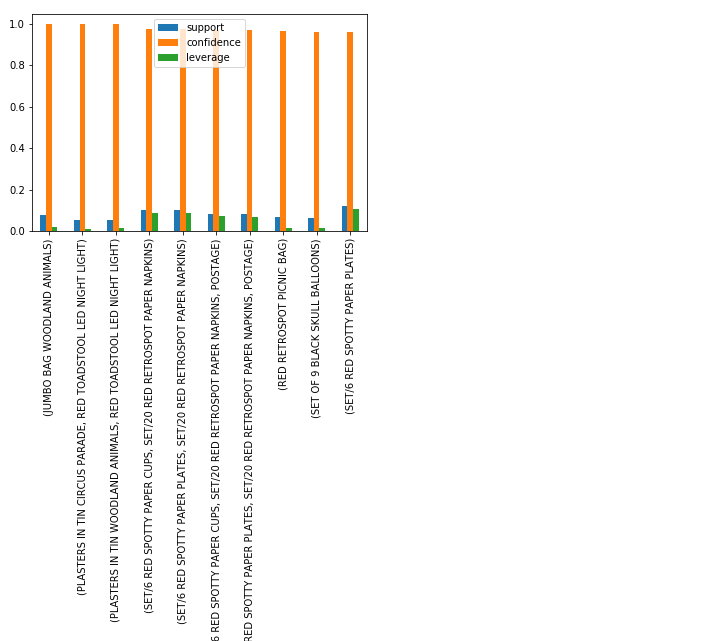
**3.4. Kết quả thu được**

Với việc phân chia dữ liệu và gom nhóm dữ liệu theo 4 quốc gia là: France, Sweden, Portuga. Chúng tôi thu được kết quả thống kê chi tiết các đơn hàng theo từng quốc gia về nội dung các mặt hàng nào được bán phổ biến theo từng quốc gia. Các mặt hàng nào thường được mua đi kèm với nhau theo từng quốc gia. Từ đó rút ra được những kiến thức xã hội và thói quen của khách hàng theo từng quốc gia kể trên.

**min\_support = 0.05**

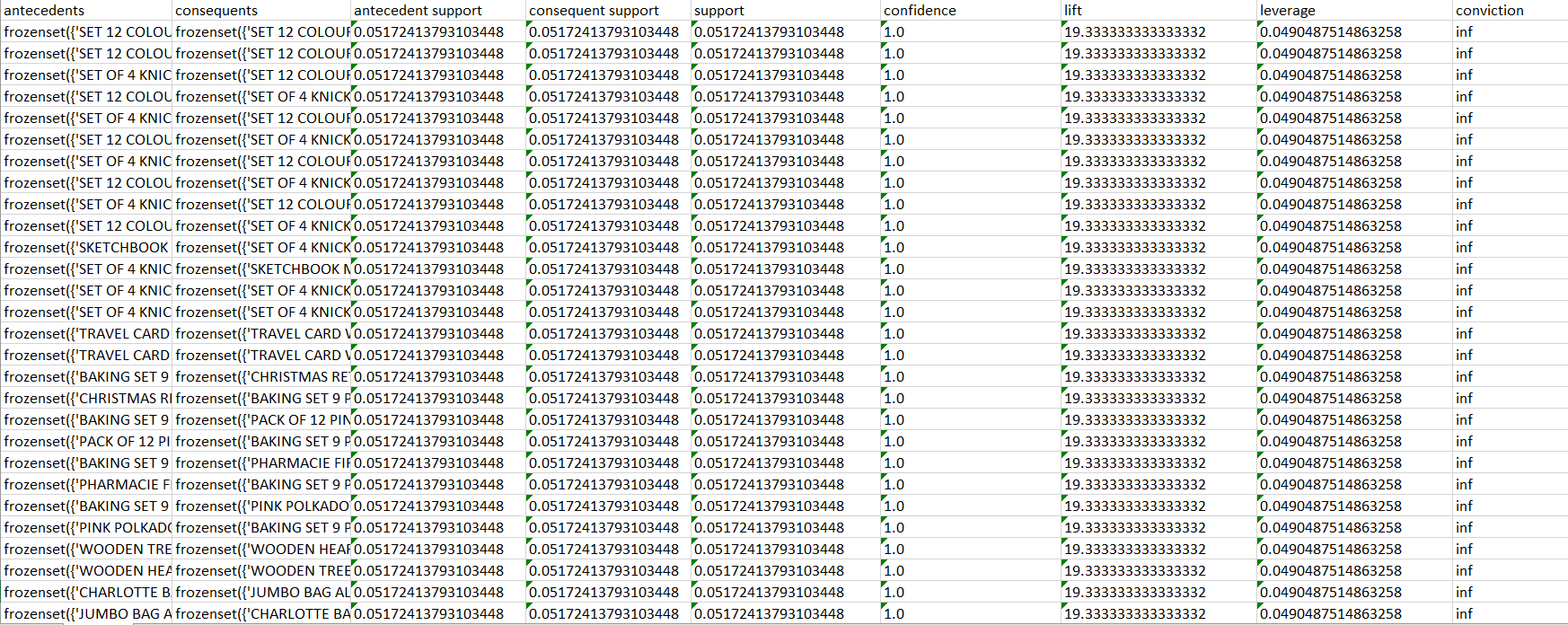
* + Kết quả thu được đối với nước France:

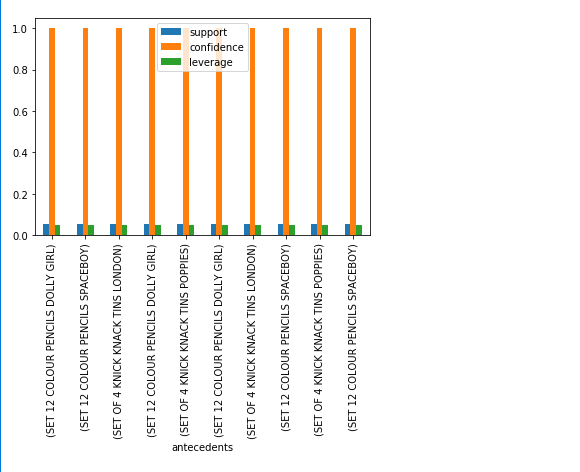




=> Từ sản lượng trên, có thể thấy ly giấy và giấy, đĩa được mua chung ở Pháp. Điều này là do người Pháp có văn hóa tụ tập với bạn bè và gia đình của họ ít nhất một lần một tuần. Ngoài ra, kể từ khi chính phủ Pháp cấm sử dụng nhựa trong nước, người dân phải mua các lựa chọn thay thế làm từ giấy.

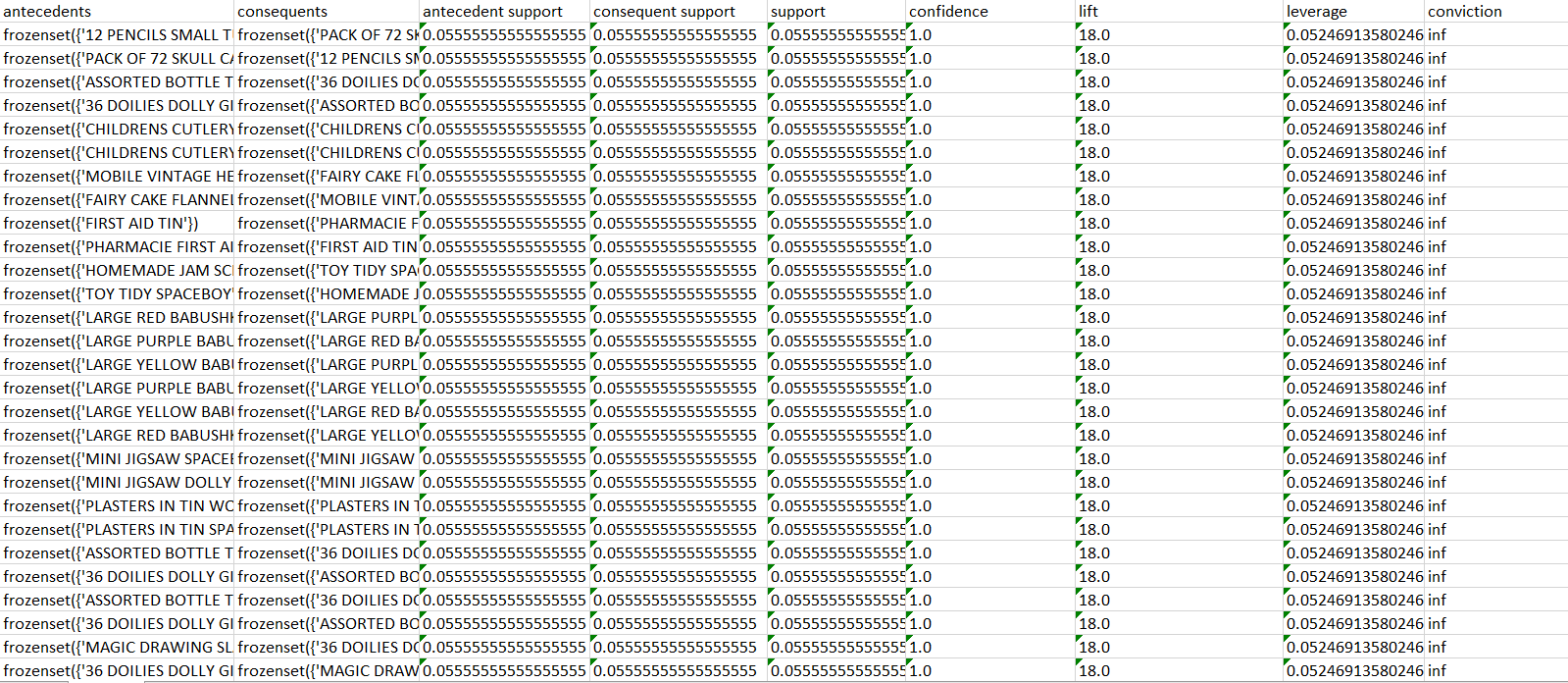
* + Kết quả thu được đối với nước Portuga:

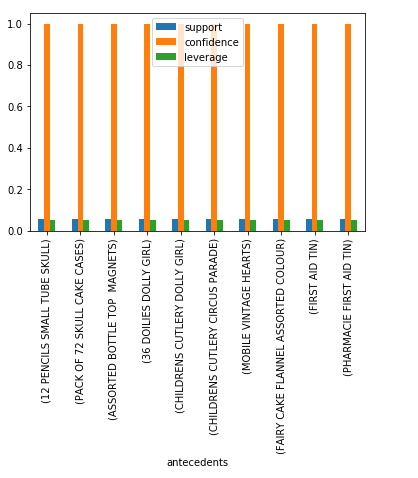




=> Khi phân tích các quy tắc kết hợp cho các giao dịch tiếng Bồ Đào Nha, người ta quan sát thấy rằng bộ Tiffin (Knick Knack Tins) và bút chì màu. Hai sản phẩm này thường dành cho trẻ đang đi học tiểu học. Hai sản phẩm này được yêu cầu bởi trẻ em ở trường để mang theo bữa ăn trưa và cho các công việc sáng tạo tương ứng và do đó, về mặt logic, chúng được ghép nối với nhau.

* + Kết quả thu được đối với nước Sweden:





=> Phân tích các quy luật trên, người ta thấy rằng dao kéo của con trai và con gái là cặp đôi với nhau. Điều này có ý nghĩa thiết thực vì khi cha mẹ đi mua sắm dao kéo cho con cái của mình, họ sẽ muốn sản phẩm được tùy chỉnh một chút theo ý muốn của đứa trẻ.

**Chương 4: Mở rộng và kết luận**

Bên cạnh thuật toán Apriori còn có thêm thuật toán FP-growth cùng được sử dụng để phục vụ cho việc khai phá các bài toán dạng kết hợp.

Thuật toán Apriori có chi phí lớn nhưng lại kém hiệu quả. Để khắc phục nhược điểm này, J. Han,

J Pei, Y. Yin và R. Mao đề xuất thuật toán FP-growth. Thuật toán FP-growth được xây dựng với 3 kỹ thuật chính:

(1) Nén dữ liệu thích hợp vào một cấu trúc cây gọi là cây FP-tree. Chỉ có các 1-tập mục (1-item) ở trong cây và các nút của cây được sắp xếp để các nút xuất hiện thường xuyên hơn có thể dễ dàng chia sẻ với các nút xuất hiện ít hơn.

(2) Thực hiện phương pháp khai phá phát triển (growth) từng đoạn dựa trên cây FP-tree gọi là phương pháp FP-growth.

(3) Kỹ thuật tìm kiếm được dùng ở đây là dựa vào sự phân chia, “chia để trị”, phân rã nhiệm vụ khai phá thành các nhiệm vụ nhỏ hơn.

**Các hướng chính mở rộng của khai phá luật kết hợp**

Lĩnh vực khai phá luật kết hợp cho đến nay đã được nghiên cứu và phát triển theo nhiều hướng khác nhau. Một trong số các hướng chính mở rộng là:

* + Luật kết hợp nhị phân
  + Luật kết hợp có thuộc tính số và thuộc tính hạng mục
  + Thuật toán kinh điển Apriori
  + Luật kết hợp tiếp cận theo hướng tập thô
  + Luật kết hợp nhiều mức
  + Luật kết hợp mờ và thêm một vài cách thức tiếp cận khác

Trong thực tế, chương trình ứng dụng khai phá luật kết hợp đã được sử dụng thành công cho nhiều lĩnh vực trong đời sống và cụ thể nhất là lĩnh vực kinh doanh bán hàng.

**Kết Luận**

Khai phá luật kết hợp là một trong những kỹ thuật quan trọng, mang tính thời sự đối với ngành công nghệ thông tin thế giới hiện nay. Sự bùng nổ về thông tin và các ứng dụng ngày càng phát triển mạnh mẽ và rộng rãi trong nhiều lĩnh vực đời sống khiến cho nhu cầu xử lý khối lượng dữ liệu khổng lồ để đề xuất ra những thông tin, tri thức hữu ích cho người dùng một cách tự động, nhanh chóng, chính xác trở thành yêu tố quan trọng cho các doanh nghiệp và cá nhân phát triển. Hiện nay, khai phá dữ liệu đang được áp dụng vào rất nhiều lĩnh vực trong đời sống và xã hội và thực tế có rất nhiều công ty đã áp dụng các mô hình, kỹ thuật khai phá dữ liệu vào trong các hoạt động sản xuất, chăm sóc,…

**Tài liệu tham khảo**

[1] https://www.geeksforgeeks.org

[2] <https://viblo.asia>

[3] <https://kipalog.com/>

[4] https://trituenhantao.info/

[5] https://tapchicongthuong.vn/