

TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

\*\*\*



**TIỂU LUẬN MÔN KHAI PHÁ DỮ LIỆU**

ĐỀ TÀI: Thuật toán Apriori trong phân tích giỏ thị trường

NHÓM 3:

* Nguyễn Tiến Dũng - 3118410057
* Lê Văn Linh - 3118410226
* Đặng Anh Quốc – 3118412047
* Trần Long Tuấn Vũ - 3118412072

# Lời cảm ơn

Lời đầu tiên em xin chân thành cám ơn các thầy cô trong khoa Công nghệ thông tin của trường đại học Sài Gòn, những người đã trực tiếp giảng dạy cung cấp kiến thức và phương pháp học tập trong 3 năm qua, đó là những nền tảng cơ bản, là những hành tranh vô cùng quý giá để em có thể bước vào sự nghiệp trong tương lai. Để có được kết quả này chúng em xin đặc biệt gửi lời cảm ơn chân thành nhất tới thầy Trịnh Tấn Đạt đã quan tâm giúp đỡ, vạch kế hoạch hướng dẫn em hoàn thành một cách tốt nhất đồ án môn học trong thời gian qua. Cuối cùng em xin chân thành cảm ơn gia đình, bạn bè đã động viên chia sẻ, giúp đỡ nhiệt tình và đóng góp nhiều ý kiến quý báu để em có thể hoàn thành đồ án ngành này.

Trong quá trình hoàn thành đồ án, vì chưa có kinh nghiêm thực tế chỉ dựa vào lý thuyết đã học, cùng với thời gian có hạn nên đồ án sẽ không tránh khỏi những sai sót. Kính mong nhận được sự góp ý, nhận xét từ thầy để kiến thức của em ngày càng hoàn thiện hơn và rút ra được nhiều kinh nghiệm bổ ích có thể áp dụng vào thực tiễn một cách hiệu quả trong tương lai.

Chúng em chân thành cám ơn!

# Lời mở đầu

Với sự phát triển của công nghệ thông tin và ứng dụng công nghệ thông tin trong nhiều lĩnh vực đời sống, kinh tế xã hội với lượng dữ liệu tích lũy ngày càng nhiều. Tuy nhiên, theo thống kê thì chỉ có một lượng nhỏ dữ liệu (khoảng 5% đến 10%) là luôn được phân tích, số còn lại sẽ tiếp tục được thu thập và để đó. Mặt khác trong môi trường cạnh tranh, người ta càng cần có nhiều thông tin để trợ giúp việc ra quyết định và trả lời dựa trên khối lượng dữ liệu khổng lồ đã có. Từ đây, các phương pháp khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế đã làm phát triển một hướng kỹ thuật mới đó là **Khai phá dữ liệu (KDD – Knowledge Discovery and Data Mining)**.

Khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau ở các nước trên thế giới, tại Việt Nam kỹ thuật này tương đối còn mới mẻ tuy nhiên cũng đang được nghiên cứu và dần đưa vào ứng dụng. Mục đích của khai phá dữ liệu chính là tìm ra các mẫu hoặc các mô hình đang tồn tại trong cơ sở dữ liệu nhưng vẫn còn bị che khuất bời hàng núi dữ liệu.

Môn học “Khai phá dữ liệu” cung cấp cho sinh viên công nghệ thông tin cái nhìn tổng quan về các thuật toán trong khai phá dữ liệu. Với những kiến thức đã được học, trong bài tiểu luận này nhóm chúng em sẽ tập trung vào thuật toán Apriori dựa vào luật khai phá kết hợp để giải quyết bài toán tìm các dữ liệu mà nhóm sẽ đề ra trong các chương tiếp theo. Luận văn bao gồm 5 chương, được bố cục như sau:

* Chương 1 – Tổng quan: Giới thiệu bài toán khai phá tập dữ liệu, các hướng tiếp cận của bài toán và đề xuất hướng tiếp cận trong luận văn.
* Chương 2 – Cơ sở lý thuyết: Trình bày cơ sở lý thuyết cơ bản về Luật khai phá kết hợp, thuật toán Apriori.
* Chương 3 – Xây dựng thuật giải: Dựa vào thuật toán Apriori để đưa ra mô hình giải quyết bài toán.
* Chương 4 – Thực nghiệm: Cài đặt chương trình thực nghiệm, đưa ra kết quả.
* Chương 5 – Kết luận: Đưa ra nhận xét và hướng phát triển trong tương lai.

Trong bài viết khó tránh khỏi những sai sót. Xin hãy gửi phản hồi cũng như những ý kiến đóng góp cho chúng tôi để đề tài này ngày một hoàn thiện hơn. Mọi góp ý xin liên hệ nhóm tác giả:

**Tác giả**

# **Mục lục**

[Lời cảm ơn 3](#_Toc103087394)

[Lời mở đầu 4](#_Toc103087395)

[Mục lục 6](#_Toc103087396)

[Danh sách các hình 8](#_Toc103087397)

[Danh sách các bảng 9](#_Toc103087398)

[Chương 1: Tổng quan 10](#_Toc103087399)

[1. Giới thiệu chung 10](#_Toc103087400)

[1.1. Nhu cầu thực tế: 10](#_Toc103087401)

[1.2. Bài toán phân tích thị giỏ thị trường 11](#_Toc103087402)

[1.3. Một số phương pháp khai phá phổ biến hiện nay 12](#_Toc103087403)

[1.4. Các ứng dụng thực tiễn 13](#_Toc103087404)

[Chương 2: Cơ sở lí thuyết 16](#_Toc103087405)

[1. Khai phá luật kết hợp 16](#_Toc103087406)

[2. Tập các mục – itemset là gì? 16](#_Toc103087407)

[2.1. Mẫu phổ biến – frequent itemset: 17](#_Toc103087408)

[2.2. Khai thác mẫu thường xuyên - Frequent Pattern Mining (FPM): 17](#_Toc103087409)

[2.3. Luật kết hợp - Association Rules 18](#_Toc103087410)

[3. Tại sao phải khai thác các mẫu phổ biến? 20](#_Toc103087411)

[4. Thuật toán Apriori - Frequent Pattern Algorithms 20](#_Toc103087412)

[Chương 3: Xây dựng thuật giải 28](#_Toc103087413)

[1. Xây dựng thuật giải: 29](#_Toc103087414)

[2. Minh họa: 31](#_Toc103087415)

[Chương 4: Thực nghiệm 34](#_Toc103087416)

[1. Import các thư viện 34](#_Toc103087417)

[2. Đọc dữ liệu từ CSDL store\_data.csv 34](#_Toc103087418)

[3. Tạo dataFrame từ list giỏ hàng 35](#_Toc103087419)

[4. Áp dụng thuật toán Apriori với min\_sup= 0.01 37](#_Toc103087420)

[5. Sử dụng hàm association\_rules() để tính confidence 38](#_Toc103087421)

[6. Kết quả top 10 sản phẩm bán chạy 40](#_Toc103087422)

[7. Kết quả top 10 sản phẩm có Confidence > 40% 41](#_Toc103087423)

[Chương 5: Kết luận 45](#_Toc103087424)

[1. Đánh giá thuật toán 45](#_Toc103087425)

[2. Ưu điểm 45](#_Toc103087426)

[3. Nhược điểm 45](#_Toc103087427)

[4. Các phương pháp cải thiện hiểu quả thuật toán Apriori: 45](#_Toc103087428)

[Tài liệu tham khảo 47](#_Toc103087429)

# Danh sách các hình

[Hình 1.0.1 Minh họa dàn các tập mục 12](file:///D:\Apriori\Khai_Pha_Du_Lieu%20_Final.docx#_Toc103084971)

[Hình 1.0.5 Giỏ hàng trong siêu thị 13](file:///D:\Apriori\Khai_Pha_Du_Lieu%20_Final.docx#_Toc103084972)

[Hình 1.0.6 Sở đồ vận hành tư vấn trực tuyến 14](#_Toc103084973)

[Hình 1.0.7 Bài toán Churn prediction 14](file:///D:\Apriori\Khai_Pha_Du_Lieu%20_Final.docx#_Toc103084974)

[Hình 1.0.8 Danh sách mua sắm của khách hàng 15](#_Toc103084975)

[Hình 2.0.1 Công thức tính độ Support và Confidence 19](#_Toc103084976)

[Hình 2.0.2 Biểu diễn các bước trong Apriori 22](#_Toc103084977)

[Hình 3.0.1 Ví dụ độ đo hỗ trợ 29](#_Toc103084978)

[Hình 3.0.2 Ví dụ về độ đo tin cậy 30](#_Toc103084979)

[Hình 3.0.3 Ví dụ về độ đo Lift 31](#_Toc103084980)

[Hình 3.0.4 Hình minh họa tập dữ liệu về các giao dịch hàng tạp hóa 31](#_Toc103084981)

[Hình 3.0.5 Các phương pháp liên kết đối với các quy tắc liên quan đến bia 32](#_Toc103084982)

[Hình 3.0.6 Độ hỗ trợ của từng mặt hàng riêng lẻ 33](#_Toc103084983)

[Hình 4.0.1 Top 10 sản phẩm bán chạy 41](#_Toc103084984)

[Hình 4.0.2 Top 10 sản phẩm có Confidence > 40% 43](#_Toc103084985)

# Danh sách các bảng

[Bảng 2‑1 23](#_Toc103084398)

[Bảng 2‑2 24](#_Toc103084399)

[Bảng 2‑3 24](#_Toc103084400)

[Bảng 2‑4 25](#_Toc103084401)

[Bảng 2‑5 26](#_Toc103084402)

[Bảng 2‑6 26](#_Toc103084403)

[Bảng 4‑1 Cơ sở dữ liệu 35](#_Toc103084404)

[Bảng 4‑2 DataFrame 36](#_Toc103084405)

[Bảng 4‑3 Tính độ Support 37](#_Toc103084406)

[Bảng 4‑4 Tính độ Confidence 40](#_Toc103084407)

# Chương 1: Tổng quan

## 1. Giới thiệu chung

### 1.1. Nhu cầu thực tế:

Khai phá dữ liệu là một trong những lĩnh vực nghiên cứu quan trọng và ngày càng phát triển với mục tiêu là tìm tất cả các tập mẫu, liên kết, tương quan hoặc cấu trúc nhân quả có độ phổ biến cao trong tập hợp các hạng mục hoặc đối tượng trong cơ sở dữ liệu giao dịch, cơ sở dữ liệu quan hệ và các kho thông tin dữ liệu khác.

Sự phát triển của công nghệ thông tin như hiện nay đang dần thể hiện rõ hơn vai trò định hướng cho ngành bán lẻ, kinh doanh sản phẩm của các doanh nghiệp. Xu thế thị trường cạnh tranh ngày càng gay gắt đòi hỏi các doanh nghiệp cần phải có những chiến lược, giải pháp của riêng mình để đáp ứng tốt hơn mong muốn của khách hàng. Các doanh nghiệp cần tìm hiểu thông tin có giá trị và chi tiết các hàng hóa để bán tốt hơn và nâng cao hiệu quả của hoạt động thị trường. Hiện nay, doanh nghiệp bán lẻ có thể thu thập các quy trình thông qua phân tích, tìm kiếm dữ liệu phù hợp nhằm cung cấp dịch vụ tốt nhất cho người tiêu dùng. Dữ liệu lớn được mô hình hóa, chọn lọc và khai phá để thu thập thông tin hữu ích cho con người. Khai phá dữ liệu là một triển vọng và là lĩnh vực cập nhật một phần của khoa học máy tính. Thuật toán Apriori là một trong những thuật toán học máy không giám sát đối với các quy tắc tìm ra luật kết hợp. Thuật toán apriori có thể được áp dụng cho tập hợp các giao dịch của các nhóm khách hàng tìm mối liên hệ giữa các sản phẩm.

Trong những năm gần đây, kỹ thuật khai phá dữ liệu và phân lớp đã được áp dụng thành công trong việc đề xuất mô hình hỗ trợ khác nhau để nâng cao chất lượng dịch vụ bán lẻ. [1][2]

### 1.2. Bài toán phân tích thị giỏ thị trường

Tác giả Eni Heni Hermaliani [1] đã sử dụng thuật toán Apriori để hỗ trợ tìm ra quy luật mua bán sản phẩm trái cây. Tác giả J. Silva [2] bằng cách sử dụng thuật toán Arpriori để khai phá quy tắc liên kết để phân khúc khách hàng trong khu vực doanh nghiệp vừa và nhỏ. Nhóm tác giả M. Kavitha và Subbaiah [3] đã sử dụng thuật toán Aprori để trích xuất sản phẩm trong cửa hàng tạp hóa.

Mục đích nghiên cứu nhằm để xác định lợi ích của thuật toán Apriori có thể giúp sự phát triển chiến lược tiếp thị, có được mô hình liên kết và xác định các sản phẩm bán chạy nhất.

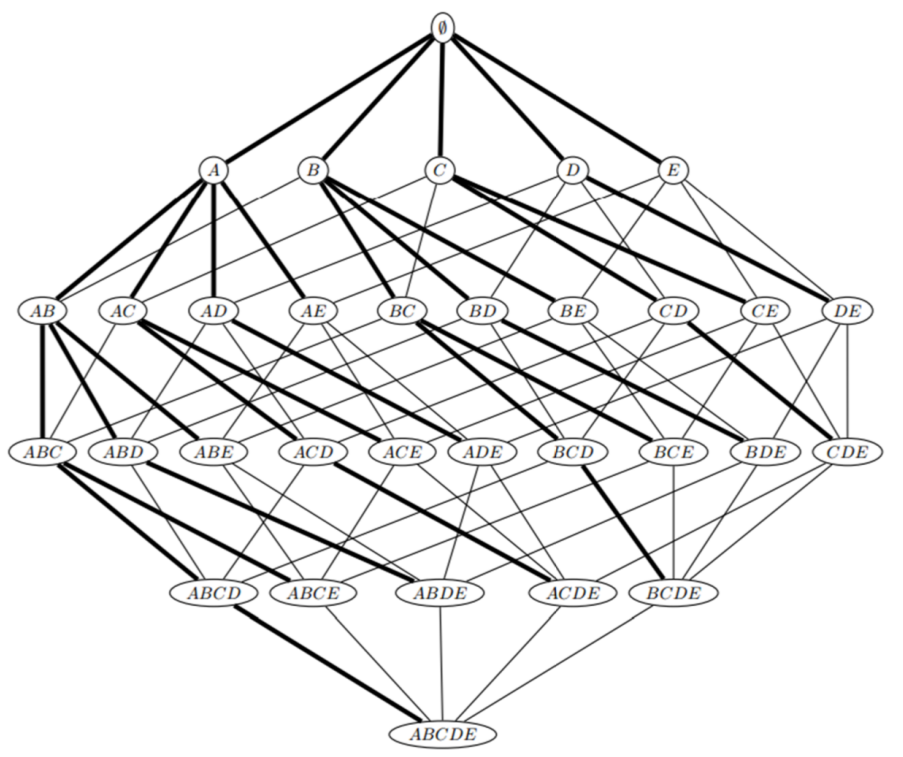
Bạn có tự hỏi tại sao trong các gian hàng siêu thị thì những món đồ cần mua lại để cạnh nhau không? Đây không phải là ngẫu nhiên mà nhằm gợi ý chúng ta sẽ mua cả hai món A và B. Từ đó ta thấy được sự hiệu quả của ứng dụng việc khai phá các tập phổ biến.

Do vai trò rất quan trọng trong việc phát triển chiến lược tiếp thị, chủ đề nghiên cứu để tìm hiểu sâu rộng về các mô hình để xác định quy luật, xác định được sản phẩm bán chạy… đã tăng nhanh trong những năm qua. Gần đây, có một vài mô hình phân lớp được nghiên cứu, đề xuất để hỗ trợ các nhà nghiên cứu trong việc xây dựng mô hình xác định quy luật, sản phẩm bán chạy [1]. Tuy nhiên, ở thời điểm hiện tại, vẫn còn thiếu các mô hình tính toán phù hợp và công cụ dự đoán với độ chính xác cao có thể hỗ trợ hiệu quả cho việc tìm kiếm luật chính xác. Bên cạnh đó, do sự tiến bộ của khoa học kỹ thuật và ảnh hưởng của cách mạng công nghiệp 4.0, dữ liệu khách hàng đã kiểm chứng thực nghiệm đang ngày càng được bổ sung nhiều hơn. Chính vì vậy, việc thiếu hụt mô hình dự đoán là một vấn đề cấp thiết cần được quan tâm giải quyết.

Tiếp tục phát triển các ý tưởng nghiên cứu trước đây, trong bài viết này nhóm chúng em tập trung vào vấn đề phân tích tìm quy luật liên kết giữa các mặt hàng trong siêu thị dựa trên dữ liệu mua hàng của khách bằng thuật toán Apriori, sử dụng ngôn ngữ Python.

### 1.3. Một số phương pháp khai phá phổ biến hiện nay

Ngoài thuật toán Apriori thì hiện nay việc khai thác tập phổ biến còn được dùng những thuật toán khác như:

* Thuật toán Brute – Force: Các tập mục có thể được duyệt theo chiều rộng (breadth first search – BFS) hoặc chiều sâu (depth first search – DFS) trên cây tiền tố.

Hình 1.0.1 Minh họa dàn các tập mục

* Thuật toán IT – Tree: có cách tiếp cận đơn giản là dựa trên phần giao nhau của tập các giao tác để tính độ phổ biến và khái niệm mới lớp tương đương nhằm chia không gian xử lý ban đầu thành tập các không gian nhỏ độc lập giúp cho việc tìm kiếm nhanh hơn. Một điểm mới nữa của phương pháp IT-tree là dựa trên phần khác nhau trên Tidset của các tập dữ liệu nhằm làm giảm kích thước bộ nhớ yêu cầu và giúp cho việc tính độ phổ biến nhanh hơn.

**\* Một số kí hiệu và định nghĩa**

1. Items I = {i1, i2, … , im}: Tập các hạng mục hay đối tượng của bài toán.
2. Itemset X: Tập các hạng mục X ⊆ I.
3. Database D: Tập hợp các giao dịch T, mỗi giao dịch là một tập danh sách các hạng mục T ⊆ I.
4. Giao dịch T chứa một itemset X: X ⊆ T.
5. Các hạng mục trong các giao dịch và các itemsets được sắp xếp theo một thứ tự nhất định:

VD: itemset 𝑋 = (𝑥1 , 𝑥2 , … , 𝑥𝑘 ), khi 𝑥1 < 𝑥2 < … < 𝑥𝑘

1. K-Itemset: itemset có độ dài k
2. Độ phổ biến hay hỗ trợ của một itemset X được định nghĩa là: support(X) = |{T ∈ D| X ⊆ T}|.
3. Độ tin cậy cậy: Cho hai tập phổ biến X và Y. Độ tin cậy của mệnh đề kéo theo X => Y là:

confidence(X=>Y) = (support (X => Y))/(support(X)).

1. Các luật tin cậy: Một luật X => Y được công nhận nếu nó thỏa mãn confidence(X=>Y) ≥ minConf.

### 1.4. Các ứng dụng thực tiễn

Các ứng dụng thực tiễn:

* Phân tích dữ liệu bán lẻ (market basket analysis).

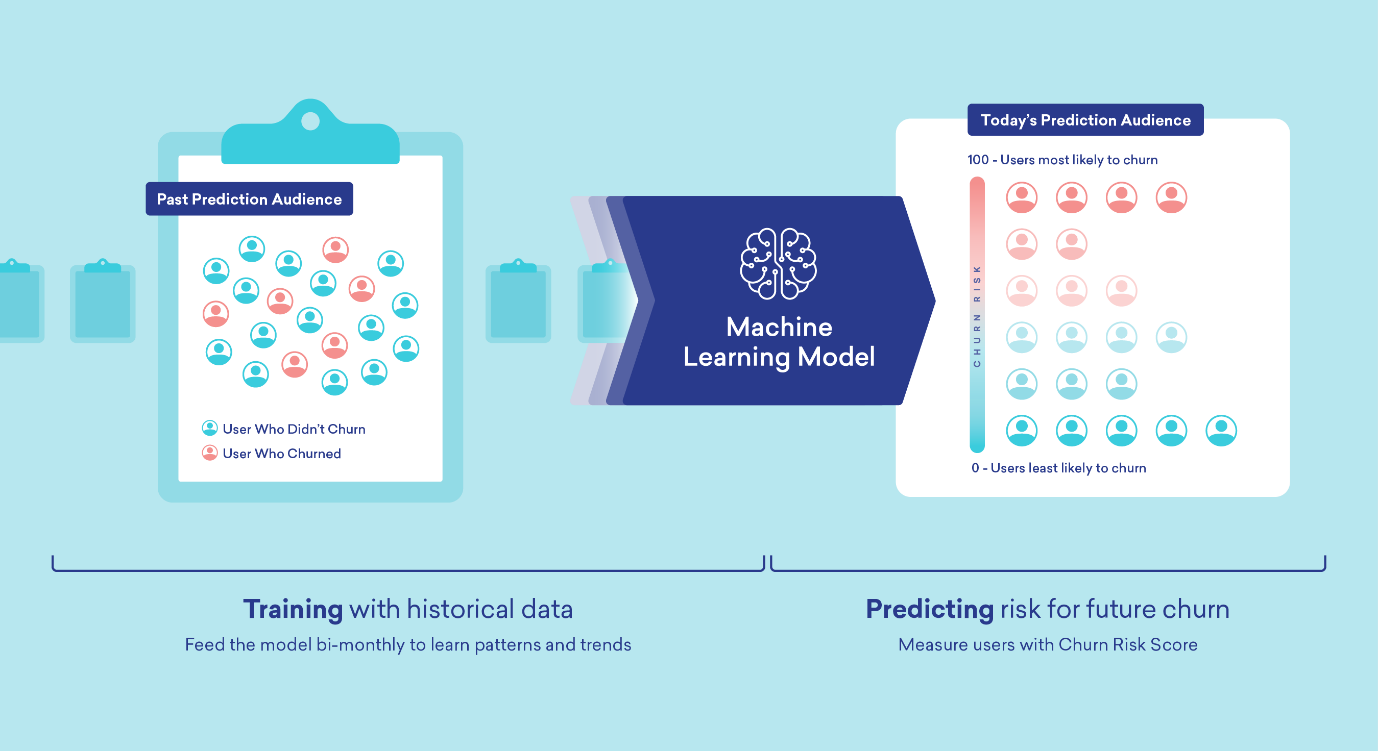
Hình 1.2 Hình minh họa siêu thị

Hình 1.0.5 Giỏ hàng trong siêu thị

* Tư vấn trực tuyến (online recommendation).



Hình 1.0.6 Sở đồ vận hành tư vấn trực tuyến

* Phân tích dữ liệu di truyền.
* Phân tích tìm ngoại lệ
* Ứng dụng trong các bài toán viễn thông (VD: churn prediction)

Hình 1.0.7 Bài toán Churn prediction

* Hiểu người dùng trực tuyến (User understanding)



Hình 1.0.8 Danh sách mua sắm của khách hàng

* Phân tích dữ liệu cấu trúc mạng.

# Chương 2: Cơ sở lí thuyết

## 1. Khai phá luật kết hợp

Khai phá luật kết hợp (Association Rule Mining) được ứng dụng phổ biến trong kinh doanh. Nó được sử dụng rộng rãi để hiểu và thử nghiệm các chiến lược kinh doanh, tiếp thị khác nhau để tăng doanh số và năng suất của các tổ chức, bao gồm chuỗi siêu thị và thị trường trực tuyến.

Khai phá luật kết hợp là học tập dựa trên quy tắc (rule-based learning) để xác định sự liên kết giữa các sản phẩm khác nhau trong cơ sở dữ liệu. Một trong những ví dụ tốt nhất và phổ biến nhất của Khai phá luật kết hợp là phân tích giỏ thị trường (Market Basket Analysis). Phân tích sự liên kết giữa các mặt hàng khác nhau có xác suất cao nhất được mua bởi khách hàng.

Các bước khai thác dữ liệu như sau:

* Data collection: đầu tiên chúng ta cần thu thập dữ liệu đáng tin cậy để dễ dàng cho việc phân tích dữ liệu sau này.
* Data cleaning: do dữ liệu phân tích rất lớn nên cần loại bỏ một số dữ liệu không cần thiết. Nếu không kết quả phân tích sẽ có sai lệch.
* Data Analysis: đây là bước phân tích data.
* Interpretation: cuối cùng, dữ liệu phân tích được có kết luận quan trọng như dự đoán.

## 2. Tập các mục – itemset là gì?

Một tập hợp các mục (**items**) cùng nhau được gọi là tập các hạng mục (**itemset**). Nếu bất kỳ tập phổ biến nào có k-item thì nó được gọi là k-itemset. Một tập hợp bao gồm hai hoặc nhiều hạng mục. Tập phổ biến xảy ra thường xuyên (được gọi là mẫu phổ biến (**frequent itemset**). Do đó, khai thác mẫu phổ biến thường xuyên là một kỹ thuật khai phá dữ liệu để xác định các hạng mục thường xảy ra cùng nhau.

VD: Bread and butter, laptop và Antivirus software, v.v.

### 2.1. Mẫu phổ biến – frequent itemset:

Một tập hợp các mục được gọi là thường xuyên nếu nó thỏa mãn một giá trị ngưỡng tối thiểu cho sự hỗ trợ (support) và độ tin cậy (confidence). Hỗ trợ hiển thị các giao dịch (transaction) với các mục được mua cùng nhau trong một giao dịch duy nhất. Độ tin cậy cho thấy các giao dịch trong đó các mục được mua lần lượt.

Đối với phương pháp khai thác tập phổ biến, chúng ta chỉ xem xét những giao dịch đáp ứng yêu cầu về độ tin cậy và hỗ trợ ngưỡng tối thiểu. Những hiểu biết sâu sắc từ các thuật toán khai phá này mang lại rất nhiều lợi ích, cắt giảm chi phí và cải thiện lợi thế cạnh tranh.

Có một khoảng thời gian cân bằng để khai thác dữ liệu và khối lượng dữ liệu để khai thác thường xuyên. Thuật toán khai phá thường xuyên là một thuật toán hiệu quả để khai phá các mẫu ẩn của các tập phổ biến trong thời gian ngắn và tiêu thụ ít bộ nhớ hơn.

### 2.2. Khai thác mẫu thường xuyên - Frequent Pattern Mining (FPM):

Thuật toán khai thác mẫu thường xuyên là một trong những kỹ thuật quan trọng nhất của khai phá dữ liệu để khám phá mối quan hệ giữa các mục khác nhau trong một tập dữ liệu (**dataset**). Các mối quan hệ này được biểu diễn dưới dạng luật kết hợp (**association rules**). Nó giúp tìm ra những bất thường trong dữ liệu.

FPM có nhiều ứng dụng trong lĩnh vực phân tích dữ liệu, lỗi phần mềm, tiếp thị chéo, phân tích chiến dịch bán hàng, phân tích rổ thị trường, v.v.

Các mục thường xuyên được phát hiện thông qua Aprori có nhiều ứng dụng trong các tác vụ khai thác dữ liệu. Các nhiệm vụ như tìm các mẫu thú vị trong cơ sở dữ liệu, tìm ra trình tự và Khai thác các luật kết hợp là quan trọng nhất trong số đó.

Luật kết hợp áp dụng cho dữ liệu giao dịch của siêu thị, tức là để kiểm tra hành vi của khách hàng về các sản phẩm đã mua. Luật kết hợp mô tả tần suất các mặt hàng được mua cùng nhau.

### 2.3. Luật kết hợp - Association Rules

Khai phá theo luật kết hợp được định nghĩa như sau:

Gọi I = {…} là một tập hợp các thuộc tính nhị phân 'n' được gọi là các mục. Đặt D = {….} Là tập hợp các giao dịch được gọi là cơ sở dữ liệu. Mỗi giao dịch trong D có một ID giao dịch duy nhất và chứa một tập hợp con của các mục trong I. Quy tắc được định nghĩa là hàm ý của dạng X-> Y trong đó X, Y? I và X? Y =?. Tập hợp các mục X và Y tương ứng được gọi là tiền trước và là hệ quả của luật.

Học các luật kết hợp được sử dụng để tìm mối quan hệ giữa các thuộc tính trong cơ sở dữ liệu lớn. Luật kết hợp, A => B, sẽ có dạng “đối với một tập hợp các giao dịch, một số giá trị của tập hợp A xác định các giá trị của tập hợp B với điều kiện đáp ứng sự hỗ trợ và độ tin cậy tối thiểu”.

**Hỗ trợ (support) và độ tin cậy (confidence) có thể được biểu diễn bằng ví dụ sau:**

VD: Bread => butter [support = 2%, confidence = 60%]

Đây là một ví dụ về luật kết hợp. Điều này có nghĩa là có 2% giao dịch đã mua bánh mì và bơ cùng nhau và có 60% khách hàng đã mua cả bánh mì và bơ.

**a. Độ hỗ trợ (Support):**

Độ hỗ trợ (Support) của luật kết hợp X =>Y là tần suất của giao dịch chứa tất cả các items trong cả hai tập X và Y. Ví dụ*, support của luật*X =>Y là 5% có nghĩa là *5% các giao dịch X và Y được mua cùng nhau.*

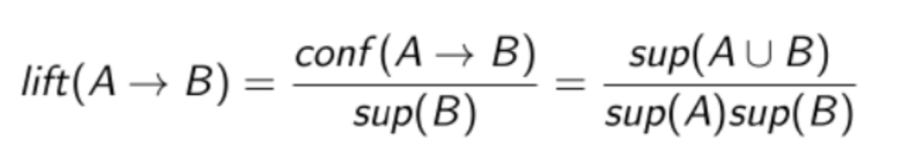
**b. Độ tin cậy (Confidence)**

Độ tin cậy (*Confidence) của luật kết hợp*X =>Y là xác suất xảy ra Y khi đã biết X. Ví dụ độ tin cậy của luật kết hợp {Apple} =>Banana} là *80% có nghĩa là 80% khách hàng mua Apple cũng mua Banana.*

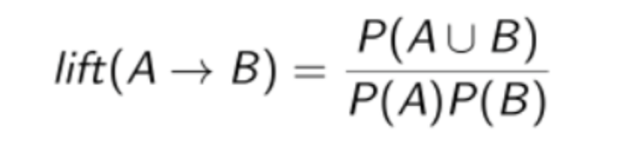
**c. Độ đo lift:**

Xét luật kết hợp A → B.

- Độ đo lift (A→B) được xác định như sau:



- Theo cách nhìn xác suất:

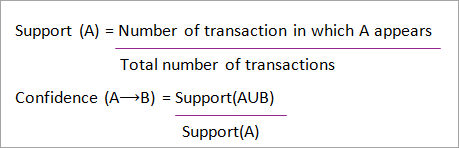


- Nếu lift(A→B) = 1: A và B độc lập, không nêên có mối quan hệ tương quan giữa A và B.

- Nếu lift(A→B) > 1: luậtA→Bcóýnghĩa (tương quan dương positive correlation).

- Nếu lift(A→B) < 1: luậtA→Bvà cả luật B→ Akhông có ý nghĩa (tương quan âm-negative correlation).

**Hỗ trợ (support) và độ tin cậy (confidence) cho tập các mục A và B được biểu diễn bằng các công thức sau:**



Hình 2.0.1 Công thức tính độ Support và Confidence

**Khai phá trong luật kết hợp bao gồm 2 bước:**

1. Tìm tất cả các tập phổ biến: theo định nghĩa thì mỗi itemset được gọi là tập mục thường xuyên nếu độ hỗ trợ của nó lớn hơn hoặc bằng min\_sup.
2. Tạo luật kết hợp mạnh từ các tập mục ở trên: theo định nghĩa thì những luật kết hợp mạnh phải có độ hỗ trợ và độ tin cậy lớn hơn min\_sup và min\_cof tương ứng.

**Khó khăn trong khai phá tập phổ biến:**

Thách thức chính trong khai phá các tập phổ biến (frequency itemsets) từ một tập dữ liệu lớn chính là việc tạo ra một lượng cực lớn các tập mục thỏa mãn độ hỗ trợ tối thiểu (min\_sup), đặc biệt khi min\_sup được cho giá trị cực nhỏ.

Điều này xảy ra bởi vì một tập mục được coi là thường xuyên nếu các tập con của nó cũng là những tập mục thường xuyên. Như vậy một tập mục dài sẽ chứa một số tổ hợp các tập mục con thường xuyên ngắn hơn.

VD: Một tập mục có chiều dài 100 – {a1, a2,… ,a100 } sẽ chứa:

* = = 100 1 – itemset thường xuyên
* =
* …
* = = 1
* Tổng hợp tổ hợp này lại + + … + =

## 3. Tại sao phải khai thác các mẫu phổ biến?

Khai thác mẫu hoặc tập phổ biến thường xuyên được sử dụng rộng rãi vì các ứng dụng đa dạng của nó trong quy tắc kết hợp khai thác (mining associations rule), tương quan và ràng buộc mẫu đồ thị dựa trên các mẫu thường xuyên, mẫu tuần tự và nhiều tác vụ khai thác dữ liệu khác.

## 4. Thuật toán Apriori - Frequent Pattern Algorithms

Thuật toán Apriori là thuật toán đầu tiên được đề xuất để khai thác tập phổ biến. Sau đó nó được cải tiến bởi R. Agarwal và R. Srikant và được biết đến với cái tên Apriori. Thuật toán này sử dụng hai bước "nối" và "tỉa" để giảm không gian tìm kiếm. Đây là một cách tiếp cận lặp đi lặp lại để khám phá các tập phổ biến nhất.

**Định nghĩa:**

Xác suất mục I không thường xuyên nếu như:

* P (I) < ngưỡng hỗ trợ tối thiểu thì I không thường xuyên.
* P (I + A) < ngưỡng hỗ trợ tối thiểu, thì I + A không thường xuyên, trong đó A cũng thuộc tập phổ biến.
* Nếu một tập hợp vật phẩm có giá trị nhỏ hơn mức hỗ trợ tối thiểu thì tất cả các tập hợp con của nó cũng sẽ giảm xuống dưới mức hỗ trợ tối thiểu và do đó có thể bị bỏ qua. Thuộc tính này được gọi là thuộc tính chống đơn điệu (Antimonotone property).

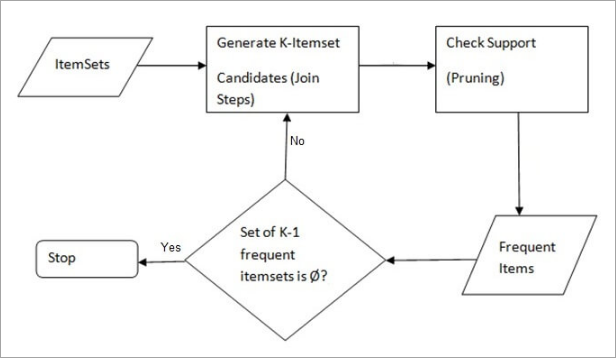
**\* Các bước trong thuật toán Apriori:**

1. **Bước nối (join step):** Bước này tạo (K + 1) tập vật phẩm từ tập vật phẩm K bằng cách ghép từng mục với chính nó.
2. **Bước tỉa (prune step**): Bước này quét số lượng của từng mục trong cơ sở dữ liệu. Nếu mục ứng cử không đáp ứng hỗ trợ tối thiểu, thì nó được coi là không thường xuyên và do đó nó bị loại bỏ. Bước này được thực hiện để giảm kích thước của các tập mục ứng viên.

**\* Các bước trong Apriori:**

Thuật toán Apriori là một chuỗi các bước cần tuân theo để tìm tập phổ biến nhất trong cơ sở dữ liệu đã cho. Kỹ thuật khai thác dữ liệu này tuân theo phép nối và các bước lược bỏ lặp đi lặp lại cho đến khi đạt được tập phổ biến nhất. Một ngưỡng hỗ trợ tối thiểu được đưa ra trong sự cố hoặc do người dùng giả định.

1. Trong lần lặp đầu tiên của thuật toán, mỗi mục được coi là ứng cử viên 1-itemsets. Thuật toán sẽ đếm số lần xuất hiện của từng mục.
2. Để có một số hỗ trợ tối thiểu, min\_sup (ví dụ: 2). Tập hợp 1 - tập phổ biến có sự xuất hiện của nó thỏa mãn điều kiện tối thiểu được xác định. Chỉ những ứng cử viên có số lượng lớn hơn hoặc bằng min\_sup, mới được đưa ra trước cho lần lặp tiếp theo và những ứng cử viên khác bị loại bỏ.
3. Tiếp theo, các mục thường xuyên của tập hợp 2 mục có min\_sup được phát hiện. Đối với điều này trong bước nối, tập hợp 2 vật phẩm được tạo bằng cách tạo thành một nhóm 2 vật phẩm bằng cách kết hợp các vật phẩm với chính nó.
4. Các ứng cử viên tập hợp 2 mục được lược bỏ bằng cách sử dụng giá trị ngưỡng min-sup. Bây giờ bảng sẽ có 2 –itemsets chỉ có min-sup.
5. Lần lặp tiếp theo sẽ tạo thành 3 –itemsets sử dụng bước nối và bước cắt tỉa. Sự lặp lại này sẽ tuân theo thuộc tính antimonotone trong đó các tập con của tập hợp 3 mục, tức là tập hợp con 2 –itemset của mỗi nhóm nằm trong min\_sup. Nếu tất cả các tập con 2-itemet là thường xuyên thì tập superset sẽ thường xuyên, nếu không nó sẽ bị lược bớt.
6. Bước tiếp theo sẽ làm theo cách tạo tập hợp 4 mục bằng cách ghép 3 mục với chính nó và cắt bớt nếu tập hợp con của nó không đáp ứng tiêu chí min\_sup. Thuật toán bị dừng khi đạt được tập phổ biến nhất.



Hình 2.0.2 Biểu diễn các bước trong Apriori

**Ví dụ: Ngưỡng hỗ trợ (Support threshold) = 50%, Độ tin cậy = 60%**

| **Transaction** | **List of items** |
| --- | --- |
| T1 | I1,I2,I3 |
| T2 | I2,I3,I4 |
| T3 | I4,I5 |
| T4 | I1,I2,I4 |
| T5 | I1,I2,I3,I5 |
| T6 | I1,I2,I3,I4 |

Bảng 2‑1

**Kết quả:**

Support threshold=50% => 0.5\*6= 3 => min\_sup=3

**1. Đếm các mục**

| **Item** | **Count** |
| --- | --- |
| I1 | 4 |
| I2 | 5 |
| I3 | 4 |
| I4 | 4 |
| I5 | 2 |

Bảng 2‑2

**2.** **Prune Step:** **BẢNG 2-2** cho thấy mục I5 không đáp ứng min\_sup = 3, do đó nó bị xóa, chỉ I1, I2, I3, I4 đáp ứng số lượng min\_sup.

| **Item** | **Count** |
| --- | --- |
| I1 | 4 |
| I2 | 5 |
| I3 | 4 |
| I4 | 4 |

Bảng 2‑3

**3.** **Join Step:** Mẫu 2-itemset. Từ BẢNG 2-1 tìm ra sự xuất hiện của 2-itemet

| **Item** | **Count** |
| --- | --- |
| I1,I2 | 4 |
| I1,I3 | 3 |
| I1,I4 | 2 |
| I2,I3 | 4 |
| I2,I4 | 3 |
| I3,I4 | 2 |

Bảng 2‑4

**4.** **Prune Step:** **BẢNG -4**cho thấy rằng tập hợp mục {I1, I4} và {I3, I4} không đáp ứng min\_sup, do đó nó bị xóa.

| **Item** | **Count** |
| --- | --- |
| I1,I2 | 4 |
| I1,I3 | 3 |
| I2,I3 | 4 |
| I2,I4 | 3 |

Bảng 2‑5

**5.** **Join and Prune Step:** Mẫu 3-itemset. Từ Bảng 2-1 tìm ra sự xuất hiện của bộ 3 mục. Từ Bảng 2-5, hãy tìm ra các tập con 2 mục hỗ trợ min\_sup.

Chúng ta có thể thấy các tập con {I1, I2, I3}, {I1, I2}, {I1, I3}, {I2, I3} đang xảy ra trong Bảng 2-5 do đó {I1, I2, I3} là thường xuyên.

Chúng ta có thể thấy các tập con {I1, I2, I4}, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} không thường xuyên, vì nó không xảy ra trong Bảng 2-5 . do đó {I1, I2, I4} không thường xuyên, do đó nó bị xóa.

| **Item** |
| --- |
| I1,I2,I3 |
| I1,I2,I4 |
| I1,I3,I4 |
| I2,I3,I4 |

Bảng 2‑6

**Chỉ {I1, I2, I3} là thường xuyên.**

**6. Áp dụng luật kết hợp: Từ tập hợp phổ biến được phát hiện ở trên, kết hợp có thể là:**

{I1, I2} => {I3}

Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4)\* 100 = 75%

{I1, I3} => {I2}

Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3)\* 100 = 100%

{I2, I3} => {I1}

Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4)\* 100 = 75%

{I1} => {I2, I3}

Confidence = support {I1, I2, I3} / support {I1} = (3/ 4)\* 100 = 75%

{I2} => {I1, I3}

Confidence = support {I1, I2, I3} / support {I2 = (3/ 5)\* 100 = 60%

{I3} => {I1, I2}

Confidence = support {I1, I2, I3} / support {I3} = (3/ 4)\* 100 = 75%

Điều này cho thấy rằng tất cả các quy tắc kết hợp ở trên là mạnh nếu ngưỡng tin cậy tối thiểu là 6.

**\* Ưu và khuyết điểm của thuật toán Apriori:**

Thuật toán Apriori tương đối đơn giản, dễ hiểu, dễ tiếp cận. Các bước nối và tỉa dễ dàng thực hiện trên các tập mục lớn trong các cơ sở dữ liệu lớn. Tuy nhiên, nó cũng đòi hỏi tính toán cao nếu các tập phổ biến rất lớn và mực độ hỗ trợ tối thiểu được giữa ở mức rất thấp. Ngoài ra, nó cần phải quét hết toàn bộ cơ sở dữ liệu.

# Chương 3: Xây dựng thuật giải

Trong chương 2, chúng tôi đã trình bày các kiến thức nền tảng về thuật toán Apriori. Sau đây, chúng tôi sẽ áp dụng các kiến thức đó để áp dụng vào bài phân tích giỏ thị trường.

Bài toán giỏ thị trường bắt đầu từ vấn đề khi chúng ta đi mua hàng tạp hóa, chúng ta thường có một danh sách chuẩn những thứ cần mua. Mỗi người mua sắm có một danh sách riêng biệt, tùy thuộc vào nhu cầu và sở thích của mỗi người. Một bà nội trợ có thể mua các nguyên liệu tốt cho sức khỏe cho bữa tối gia đình, trong khi một công nhân có thể mua bia và khoai tây chiên. Hiểu được các mô hình mua hàng này có thể giúp tăng doanh số bán hàng theo nhiều cách. Nếu có một cặp mặt hàng, X và Y, thường được mua cùng nhau thì:

* Cả X và Y đều có thể được đặt trên cùng một giá để người mua một mặt hàng sẽ được nhắc mua mặt hàng kia.
* Giảm giá khuyến mại chỉ có thể được áp dụng cho một trong hai mặt hàng.
* Quảng cáo trên X có thể được nhắm mục tiêu đến những người mua Y.
* X và Y có thể được kết hợp thành một sản phẩm mới, chẳng hạn như có Y trong các hương vị của X.

Mặc dù chúng ta có thể biết rằng một số mặt hàng nhất định thường được mua cùng nhau, nhưng câu hỏi đặt ra là, làm thế nào để chúng ta phát hiện ra những mối liên quan này?

Bên cạnh việc tăng lợi nhuận bán hàng, các quy tắc liên kết cũng có thể được sử dụng trong các lĩnh vực khác. Ví dụ, trong chẩn đoán y tế, hiểu được những triệu chứng nào có xu hướng mắc cùng bệnh có thể giúp cải thiện việc chăm sóc bệnh nhân và kê đơn thuốc.

## 1. Xây dựng thuật giải:

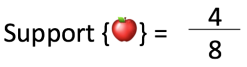
Ở chương 2, chúng tôi đã trình bày các kiến thức nền tảng về thuật toán Apriori. Trong phần này chúng tôi xin nhắc lại định nghĩa cũng như các thông số mà Apriori sử dụng để phân tích giỏ thị trường.

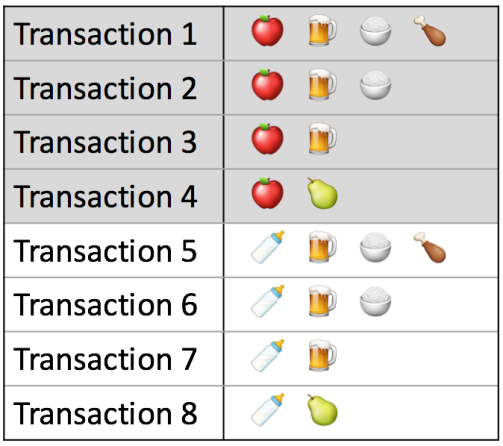
**a. Định nghĩa:**

Phân tích luật kết hợp là một kỹ thuật để khám phá cách các mục được liên kết với nhau. Có ba cách phổ biến để đo lường sự liên kết.

**b. Cách đo thứ nhất: Độ hỗ trợ:**

Điều này cho biết mức độ phổ biến của tập các mục (itemset), được đo lường bằng tỷ lệ giao dịch trong đó tập các mục xuất hiện. Trong hình 3.1.1.a bên dưới, mức hỗ trợ của {apple} là 4 trên 8 hoặc 50%. Các tập vật phẩm cũng có thể chứa nhiều mục. Ví dụ: độ ủng hộ của {apple, beer, rice} là 2 trên 8 hoặc 25%.



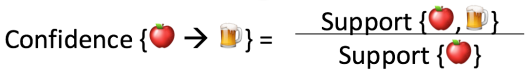


Hình 3.0.1 Ví dụ độ đo hỗ trợ

Nếu phát hiện ra rằng doanh số bán các mặt hàng vượt quá một tỷ lệ nhất định có xu hướng ảnh hưởng đáng kể đến lợi nhuận, thì có thể cân nhắc sử dụng tỷ lệ đó làm ngưỡng hỗ trợ của mình. Sau đó, xác định các tập vật phẩm có giá trị hỗ trợ trên ngưỡng này là tập hợp vật phẩm quan trọng.

**c. Cách đo thứ hai: Độ tin cậy:**

Độ tin cậy cho biết khả mặt hàng Y được mua khi mặt hàng X được mua, được biểu thị bằng {X -> Y}. Điều này được đo lường bằng tỷ lệ giao dịch với mặt hàng X, trong đó mặt hàng Y cũng xuất hiện. Trong Bảng 1, độ tin cậy của {apple -> beer} là 3 trên 4 hoặc 75%.

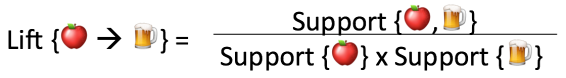


Hình 3.0.2 Ví dụ về độ đo tin cậy

Một nhược điểm của độ đo tin cậy đó là nó có thể miêu tả sai tầm quan trọng của một liên kết. Điều này là do nó chỉ giải thích cho mức độ phổ biến của táo, chứ không phải các loại bia. Nếu các loại bia nói chung cũng rất phổ biến, sẽ có khả năng cao hơn là một giao dịch có táo cũng sẽ chứa bia, do đó sẽ làm tăng độ tin cậy. Để tính đến mức độ phổ biến cơ bản của cả hai mặt hàng cấu thành, chúng tôi sử dụng thước đo thứ ba gọi là mức tăng.

**d. Cách đo thứ ba: Độ đo Lift**

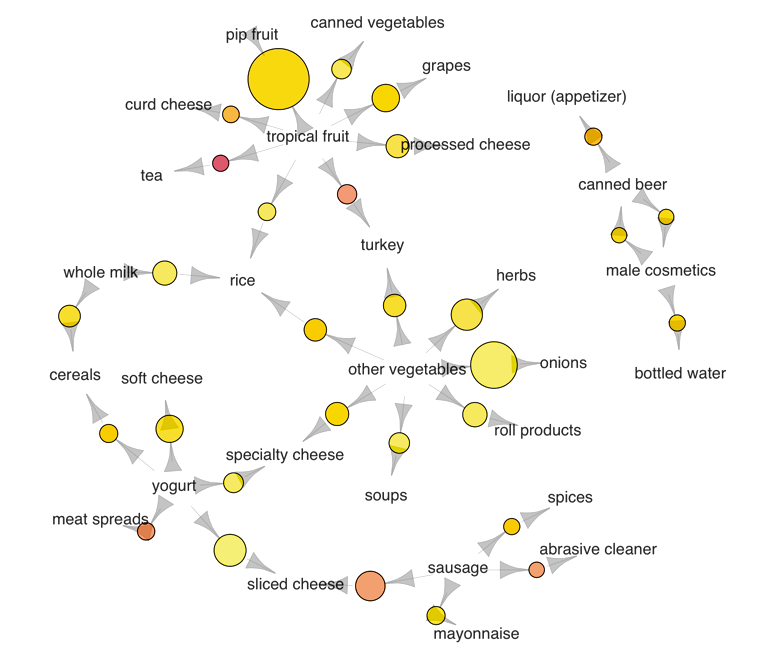
Độ đo lift cho biết khả năng mặt hàng Y được mua khi mặt hàng X được mua, đồng thời kiểm soát mức độ phổ biến của mặt hàng Y. Trong Bảng 1, mức tăng của {apple -> beer} là 1, nghĩa là không có sự liên kết giữa các mục. Giá trị nâng cao hơn 1 có nghĩa là mặt hàng Y có khả năng được mua nếu mua mặt hàng X, trong khi giá trị nhỏ hơn 1 có nghĩa là mặt hàng Y khó có thể được mua nếu mặt hàng X được mua.



Hình 3.0.3 Ví dụ về độ đo Lift

## 2. Minh họa:

Chúng tôi sử dụng tập dữ liệu về các giao dịch hàng tạp hóa từ thư viện arules R. Nó chứa các giao dịch thực tế tại một cửa hàng tạp hóa trong 30 ngày. Biểu đồ mạng bên dưới hiển thị các liên kết giữa các mục đã chọn. Vòng tròn lớn hơn ngụ ý mức hỗ trợ cao hơn, trong khi vòng tròn màu đỏ ngụ ý mức nâng cao hơn:



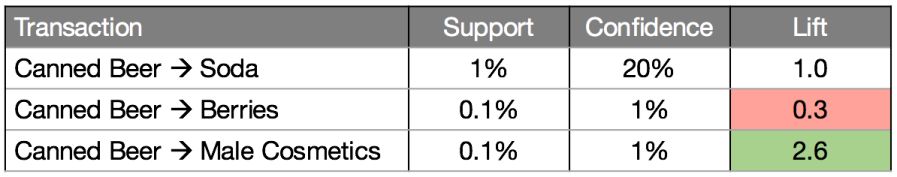
Hình 3.0.4 Hình minh họa tập dữ liệu về các giao dịch hàng tạp hóa

Liên kết giữa các mục đã chọn. Trực quan hóa bằng cách sử dụng thư viện arulesViz R.

Chúng ta có thể quan sát thấy được có một số mô hình mua hàng. Ví dụ:

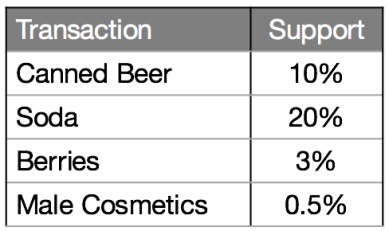
* Giao dịch phổ biến nhất là pip và trái cây nhiệt đới
* Một giao dịch phổ biến khác là hành và các loại rau khác
* Nếu ai đó mua thịt phết, rất có thể người đó cũng đã mua sữa chua
* Tương đối nhiều người mua xúc xích cùng với pho mát cắt lát
* Nếu ai đó mua trà, rất có thể anh ta cũng đã mua trái cây, có thể tạo cảm hứng cho việc sản xuất trà có hương vị trái cây

Chúng ta hãy nhớ lại một nhược điểm của thước đo độ tin cậy là nó có xu hướng trình bày sai về tầm quan trọng của một liên kết. Để chứng minh điều này, chúng tôi quay lại tập dữ liệu chính để chọn 3 quy tắc kết hợp có chứa bia:



Hình 3.0.5 Các phương pháp liên kết đối với các quy tắc liên quan đến bia

Quy tắc {beer -> soda} có độ tin cậy cao nhất là 20%. Tuy nhiên, cả bia và nước ngọt đều xuất hiện thường xuyên trong tất cả các giao dịch (xem Bảng 3), vì vậy sự liên kết của chúng có thể chỉ đơn giản là một sự may rủi. Điều này được xác nhận bởi giá trị nâng của {beer -> soda}, là 1, ngụ ý không có mối liên hệ nào giữa bia và soda.



Hình 3.0.6 Độ hỗ trợ của từng mặt hàng riêng lẻ

Mặt khác, quy tắc {bia -> mỹ phẩm nam} có độ tin cậy thấp, do ít mua mỹ phẩm nam nói chung. Tuy nhiên, bất cứ khi nào ai đó mua mỹ phẩm nam, anh ta cũng rất có thể mua bia, được suy ra từ giá trị nâng cao là 2,6. Điều ngược lại đúng với {beer -> berries}. Với giá trị nâng dưới 1, chúng tôi có thể kết luận rằng nếu ai đó mua quả mọng, anh ta sẽ có ác cảm với bia.

Có thể dễ dàng tính toán mức độ phổ biến của một tập hợp vật phẩm, chẳng hạn như {beer, soda}. Tuy nhiên, chủ doanh nghiệp thường không hỏi về các tập hợp vật phẩm riêng lẻ. Thay vào đó, chủ sở hữu sẽ quan tâm hơn đến việc có một danh sách đầy đủ các bộ vật phẩm phổ biến. Để có được danh sách này, người ta cần tính toán các giá trị hỗ trợ cho mọi cấu hình có thể có của các mục, sau đó liệt kê các tập hợp mục đáp ứng ngưỡng hỗ trợ tối thiểu.

Trong một cửa hàng chỉ có 10 mặt hàng, tổng số cấu hình có thể có để kiểm tra sẽ là con số khổng lồ 1023. Con số này tăng theo cấp số nhân trong một cửa hàng có hàng trăm mặt hàng.

# Chương 4: Thực nghiệm

Chúng tôi cài đặt chương trình ứng dụng bằng Python version 3.10.4. Chúng tôi sử dụng Python vì Python có sẵn những thư viện như Apriori, Association\_rules và các hàm hỗ trợ rất tốt trong việc xử lý dữ liệu.

## 1. Import các thư viện

from builtins import print  
  
import pandas as pd  
from mlxtend.preprocessing import TransactionEncoder  
from mlxtend.frequent\_patterns import apriori, association\_rules  
import matplotlib.pyplot as plt  
import seaborn as sns

## 2. Đọc dữ liệu từ CSDL store\_data.csv

data = pd.read\_csv('store\_data.csv', low\_memory=False, sep='\t', header=None, names=['products'])  
lg = len(data)   
  
ls = list(data["products"].apply(lambda x: x.split(',')))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| … | … | … | … | … |
| burgers | meatballs | eggs |  |  |
| chutney |  |  |  |  |
| turkey | avocado |  |  |  |
| mineral water | milk | energy bar | whole wheat rice | green tea |
| low fat yogurt |  |  |  |  |
| whole wheat pasta | french fries |  |  |  |
| soup | light cream | shallot |  |  |
| frozen vegetables | spaghetti | green tea |  |  |
| french fries |  |  |  |  |
| eggs | pet food |  |  |  |
| … | … | … | … | … |

Bảng 4‑1 Cơ sở dữ liệu

## 3. Tạo dataFrame từ list giỏ hàng

te = TransactionEncoder()  
te\_data = te.fit(ls).transform(ls)  
dataf = pd.DataFrame(te\_data, columns=te.columns\_)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | asparagus | almonds | antioxydant juice | asparagus | avocado | babies food | bacon | barbecue sauce | black  tea | … |
| 0 | False | True | True | False | True | False | False | False | False | … |
| 1 | False | False | False | False | False | False | False | False | False | … |
| 2 | False | False | False | False | False | False | False | False | False | … |
| 3 | False | False | False | False | True | False | False | False | False | … |
| 4 | False | False | False | False | False | False | False | False | False | … |
| 5 | False | False | False | False | False | False | False | False | False | … |
| 6 | False | False | False | False | False | False | False | False | False | … |
| 7 | False | False | False | False | False | False | False | False | False | … |
| 8 | False | False | False | False | False | False | False | False | False | … |
| 9 | False | False | False | False | False | False | False | False | False | … |
| 10 | … | … | … | … | … | … | … | … | … | … |

Bảng 4‑2 DataFrame

## 4. Áp dụng thuật toán Apriori với min\_sup= 0.01

df = apriori(dataf, min\_support=0.01, use\_colnames=True)

|  |  |  |
| --- | --- | --- |
|  | support | itemsets |
| 0 | 0.020397280362618318 | 'almonds' |
| 1 | 0.03332888948140248 | 'avocado' |
| 2 | 0.010798560191974404 | 'barbecue sauce' |
| 3 | 0.014264764698040262 | 'black tea' |
| 4 | 0.011465137981602452 | 'body spray' |
| 5 | 0.03372883615517931 | 'brownies' |
| 6 | 0.0871883748833489 | 'burgers' |
| 7 | 0.030129316091187842 | 'butter' |
| 8 | 0.08105585921877083 | 'cake' |
| 9 | 0.01533128916144514 | 'carrots' |
| 10 | 0.025729902679642713 | 'cereals' |
| 11 | 0.04679376083188908 | 'champagne' |
| 12 | 0.05999200106652446 | 'chicken' |
| 13 | 0.1638448206905746 | 'chocolate' |

Bảng 4‑3 Tính độ Support

## 5. Sử dụng hàm association\_rules() để tính confidence

df\_ar = association\_rules(df, metric="confidence", min\_threshold = 0.4)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | antecedents | consequents | antecedent support | consequent support | support | confidence | lift | leverage | conviction |
| 0 | 'ground beef' | 'mineral water' | 0.09825356619117451 | 0.23836821757099053 | 0.040927876283162246 | 0.41655359565807326 | 1.7475215442008991 | 0.01750734884017864 | 1.305401450349256 |
| 1 | 'olive oil' | 'mineral water' | 0.0658578856152513 | 0.23836821757099053 | 0.027596320490601255 | 0.4190283400809717 | 1.7579035676439423 | 0.011897893683499625 | 1.3109620227175287 |
| 2 | 'salmon' | 'mineral water' | 0.04252766297826956 | 0.23836821757099053 | 0.017064391414478068 | 0.4012539184952978 | 1.6833364891684726 | 0.006927148192888149 | 1.2720447046850996 |
| 3 | 'soup' | 'mineral water' | 0.05052659645380616 | 0.23836821757099053 | 0.023063591521130515 | 0.45646437994722955 | 1.9149548735929356 | 0.01101965678450801 | 1.4012545900028865 |
| 4 | 'chocolate', 'eggs' | 'mineral water' | 0.03319557392347687 | 0.23836821757099053 | 0.013464871350486601 | 0.40562248995983935 | 1.7016634771749188 | 0.005552101563101368 | 1.2813940123298875 |
| 5 | 'ground beef', 'chocolate' | 'mineral water' | 0.023063591521130515 | 0.23836821757099053 | 0.010931875749900012 | 0.47398843930635837 | 1.988471634920019 | 0.005434248548222722 | 1.4479373446177872 |
| 6 | 'chocolate', 'milk' | 'mineral water' | 0.03212904946007199 | 0.23836821757099053 | 0.013998133582189041 | 0.43568464730290457 | 1.827779943746693 | 0.006339589330141486 | 1.3496563203337595 |
| 7 | 'spaghetti', 'chocolate' | 'mineral water' | 0.03919477403012932 | 0.23836821757099053 | 0.01586455139314758 | 0.40476190476190477 | 1.6980531586236285 | 0.006521762969487907 | 1.2795413944807361 |
| 8 | 'ground beef', 'eggs' | 'mineral water' | 0.019997333688841486 | 0.23836821757099053 | 0.010131982402346354 | 0.5066666666666667 | 2.125563012677107 | 0.005365253614764888 | 1.5438482076263709 |
| 9 | 'eggs', 'milk' | 'mineral water' | 0.03079589388081589 | 0.23836821757099053 | 0.013064924676709772 | 0.42424242424242425 | 1.7797776421937497 | 0.005724162343834314 | 1.3228341484293324 |

Bảng 4‑4 Tính độ Confidence

## 6. Kết quả top 10 sản phẩm bán chạy

df = df.head(10)  
df['sup'] = df['support'].apply(lambda x: x \* lg)  
  
products10 = list(df['itemsets'].apply(lambda x: list(x)[0]).astype("unicode"))  
support10 = list(df['sup'])  
df10 = pd.DataFrame(list(zip(products10, support10)), columns=['products', 'support'])  
df10 = df10.reset\_index()

plt.figure(figsize=(15, 10))  
plots = sns.barplot(x="products", y="support", data=df10)  
for bar in plots.patches:  
plots.annotate(int(bar.get\_height()),  
 (bar.get\_x() + bar.get\_width() / 2,  
 bar.get\_height()), ha='center', va='center',  
 size=15, xytext=(0, 8),  
 textcoords='offset points')  
plt.xticks(rotation=10)  
plt.xlabel("Product", size=14)  
plt.ylabel("Count of product", size=14)  
plt.title('Top 10 product purchased by customers', color='green', size=20)  
plt.show()

Chart, bar chart

Description automatically generated

Hình 4.0.1 Top 10 sản phẩm bán chạy

## 7. Kết quả top 10 sản phẩm có Confidence > 40%

def nameProduct(itemsets):  
 df\_ar1 = df\_ar.loc[df\_ar['antecedents'].apply(lambda x: x == itemsets)]  
 consequents = df\_ar1.consequents.values[0]  
 lsCon = list(consequents)  
 ls = list(itemsets)  
 s = ''  
 for x in ls[:]:  
 s += x + ', '  
 s = s[0:len(s) - 2]  
 s += ' → '  
 for x in lsCon[:]:  
 s += x + ', '  
 s = s[0:len(s) - 2]  
 return s

df\_ar.sort\_values(by="confidence", ascending=False, inplace=True)  
products=list(

df\_ar['antecedents'].head(10).apply(

lambda x:nameProduct(x)).astype("unicode"))  
confidence =list(df\_ar['confidence'].head(10).values)  
support = list(df\_ar['support'].head(10).apply(lambda x: x \* lg))  
df1 = pd.DataFrame(list(zip(products, support, confidence)), columns=['products', 'support', 'confidence'])  
df1 = df1.reset\_index()

plt.figure(figsize=(15, 10))  
plots = sns.barplot(x="products", y="confidence", data=df1, palette='gnuplot')  
i = 0  
for bar in plots.patches:  
 plots.annotate(int(support[i]),  
 (bar.get\_x() + bar.get\_width() / 2,  
 bar.get\_height()), ha='center', va='center',  
 size=15, xytext=(0, 8),  
 textcoords='offset points')  
 i += 1  
plt.xticks(rotation=6)  
plt.xlabel("Items", size=14)  
plt.ylabel("%", size=14)  
plt.ylim(0, 1)  
plt.title('Confidence', color='green', size=20)  
plt.show()

Chart, bar chart

Description automatically generated

Hình 4.0.2 Top 10 sản phẩm có Confidence > 40%

# Chương 5: Kết luận

## 1. Đánh giá thuật toán

Việc thuật toán Apriori có thể làm là dựa vào dữ liệu có sẵn để khẳng định việc gì đó xảy ra thì có tỉ lệ bao nhiêu phần trăm dữ liệu tiếp theo sẽ xảy ra. Việc này giống như là nhìn vào quá khứ để dự đoán tương lại, điều này rất có lợi cho các nhà kinh doanh.

## 2. Ưu điểm

* Đây là thuật toán đơn giản nhất trong số các phương pháp khai phá luật kết hợp.
* Các quy tắc kết quả là trực quan và dễ dàng để giao tiếp với người dùng cuối.
* Nó không yêu cầu dữ liệu được dán nhãn vì nó là thuật toán không giám sát, do đó bạn có thể sử dụng nó trong nhiều tình huống khác nhau vì dữ liệu không được gán nhãn thường dễ truy cập hơn.
* Nhiều tiện ích mở rộng đã được đề xuất trong thuật toán này

## 3. Nhược điểm

* Tốn thời gian cho việc duyệt cơ sở dữ liệu nếu I = i1, i2, .. i1000 thì số lần duyệt sẽ là 1000.
* Việc thực hiện tính độ phổ biến sẽ nhiều hơn, đơn điệu.
* Số lượng tập ứng viên lớn.

## 4. Các phương pháp cải thiện hiểu quả thuật toán Apriori:

Có nhiều phương pháp có sẵn để cải thiện hiệu quả của thuật toán:

* **Kỹ thuật dựa trên băm (Hash-Based Technique):** Phương pháp này sử dụng cấu trúc dựa trên băm được gọi là bảng băm để tạo k-itemets và số lượng tương ứng của nó. Nó sử dụng một hàm băm để tạo bảng.
* **Giảm giao dịch (Transaction Reduction):** Phương pháp này giảm số lần quét giao dịch trong các lần lặp lại. Các giao dịch không chứa các mục thường xuyên được đánh dấu hoặc loại bỏ.
* **Phân vùng (Partitioning):** Phương pháp này chỉ yêu cầu hai lần quét cơ sở dữ liệu để khai thác các tập phổ biến. Nó nói rằng đối với bất kỳ tập hợp mục nào có khả năng xuất hiện thường xuyên trong cơ sở dữ liệu, thì tập hợp đó phải xuất hiện thường xuyên trong ít nhất một trong các phân vùng của cơ sở dữ liệu.
* **Lấy mẫu (Sampling):** Phương pháp này chọn một mẫu ngẫu nhiên S từ Cơ sở dữ liệu D và sau đó tìm kiếm tập phổ biến thường xuyên trong S. Có thể mất tập phổ biến chung. Điều này có thể được giảm bớt bằng cách giảm min\_sup.
* **Đếm tập mục phẩm động (Dynamic Itemset Counting):** Kỹ thuật này có thể thêm tập hợp vật phẩm ứng viên mới tại bất kỳ điểm bắt đầu được đánh dấu nào của cơ sở dữ liệu trong quá trình quét cơ sở dữ liệu.

# Tài liệu tham khảo

[1]. E. H. Hermaliani et al, “Data Mining Technique to Determine the Pattern of Fruits Sales & Supplies Using Apriori Algorithm,” Journal of Physics: conference series, vol. 1641, 2020, Art. no. 012070.

[2]. J. Silva et al, “Association Rules Extraction for Customer Segmentation in the SMEs Sector Using the Apriori Algorithm,” International Workshop on Web Search and Data Mining (WSDM), April 29 - May 02, 2019, Leuven, Belgium.

[3]. M. Kavitha and S. Subbaiah, “Association Rule Mining using Apriori Algorithm for Extracting Product Sales Patterns in Groceries,” Int. J. Eng. Res. Technol., vol. 08, no. 03, pp. 1-4, 2020.

[4]. “Apriori Algorithm In Data Mining: Implementation With Examples”. Softwaretestinghelp, 2022, <https://www.softwaretestinghelp.com/apriori-algorithm/>. Accessed 20 Apr 2022

[5]. "AprioriAlgorithm - Geeksforgeeks". Geeksforgeeks, 2018, https://www.geeksforgeeks.org/apriori-algorithm/. Accessed 22 Apr 2022

[6]. Ong, Hong, and Xem Ong. "Khai Thác Tập Phổ Biến (Frequent Itemsets) Với Thuật Toán Apriori". Ông Xuân Hồng, 2015, https://ongxuanhong.wordpress.com/2015/08/23/khai-thac-luat-tap-pho-bien-frequent-itemsets-voi-thuat-toan-apriori/. Accessed 22 Apr 2022.

[7]. Ng, Annalyn. "Association Rules And The Apriori Algorithm: A Tutorial - Kdnuggets". *Kdnuggets*, 2022, https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html?fbclid=IwAR1adKGGvvzmyqIWLrKWXv59CpOXnwBqiv8Vc7SgsIyURoe7MO9OI42NEAM. Accessed 9 May 2022.