**TRƯỜNG ĐẠI HỌC MỞ HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

----------------------



**BÁO CÁO BÀI TẬP LỚN**

**MÔN: NHẬP MÔN KHAI PHÁ DỮ LIỆU VÀ HỌC MÁY**

**ĐỀ TÀI: NGHIÊN CỨU VÀ CÀI ĐẶT CHƯƠNG TRÌNH PHÁT HIỆN CÁC TẬP MẪU PHỔ BIẾN SỬ DỤNG THUẬT TOÁN FP-GROWTH**

**Giảng viên hướng dẫn**:  **TS.Lưu Minh Tuấn**

**Nhóm sinh viên thực hiện**: Dương Thị Hằng - 2210A03

Trần Thị Huyền - 2210A03

Lý Thị Thảo - 2110A04

Trần Thị Mai Thi - 2210A03

Nguyễn Thị Minh Thúy - 2210A02

**Hà Nội – 2024**

**DANH SÁCH CÁC THÀNH VIÊN TRONG NHÓM**

|  |  |  |
| --- | --- | --- |
| **Mã sinh viên** | **Họ và tên** | **Lớp** |
| 22A1001D0107 | Dương Thị Hằng | 2210A03 |
| 22A1001D0164 | Trần Thị Huyền | 2210A03 |
| 21A100100347 | Lý Thị Thảo | 2110A04 |
| 22A1001D0304 | Trần Thị Mai Thi | 2210A03 |
| 22A1001D0316 | Nguyễn Thị Minh Thúy | 2210A02 |

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI 2](#_Toc181305290)

[1.1. Phát biểu đề tài 2](#_Toc181305291)

[1.2. Tính cấp thiết của đề tài 2](#_Toc181305292)

[1.3. Phạm vi nghiên cứu 3](#_Toc181305293)

[1.4. Đối tượng và mục đích nghiên cứu 3](#_Toc181305294)

[1.5. Phương pháp nghiên cứu 3](#_Toc181305295)

[1.6. Cấu trúc của đề tài 4](#_Toc181305296)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 6](#_Toc181305297)

[2.1. Khái niệm về khai phá dữ liệu 6](#_Toc181305298)

[2.2. Khai phá tập mẫu phổ biến 6](#_Toc181305299)

[2.3. Thuật toán FP-Growth 7](#_Toc181305300)

[2.4. Cải tiến thuật toán FP-Growth 9](#_Toc181305301)

[2.5. Ưu và nhược điểm của thuật toán FP-Growth 9](#_Toc181305302)

[2.6. Bài toán xây dựng cấu trúc FP-Tree 10](#_Toc181305303)

[CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM 13](#_Toc181305304)

[3.1. Môi trường cài đặt 13](#_Toc181305305)

[3.2. Bộ dữ liệu 14](#_Toc181305306)

[CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ NGHIÊN CỨU 19](#_Toc181305307)

[4.1. Bảng so sánh thuật toán Apriori và FP-Growth 19](#_Toc181305308)

[4.2. Đánh giá kết quả nghiên cứu 19](#_Toc181305309)

[4.3. Xây dựng và tích hợp ứng dụng AI trên model 21](#_Toc181305310)

[5. Ứng dụng thực tiễn 22](#_Toc181305311)

[CÁC TÀI LIỆU THAM KHẢO 24](#_Toc181305312)

## **LỜI NÓI ĐẦU**

Trong thời đại bùng nổ thông tin, dữ liệu phát sinh từ các hoạt động mua sắm hàng ngày của khách hàng là một nguồn tài nguyên quý giá, cung cấp nhiều thông tin quan trọng giúp các doanh nghiệp hiểu rõ hơn về thói quen và xu hướng tiêu dùng. Khai phá dữ liệu, đặc biệt là khai phá tập mẫu phổ biến, đã trở thành một công cụ thiết yếu hỗ trợ các doanh nghiệp tối ưu hóa danh mục sản phẩm, cải thiện chất lượng dịch vụ, và đưa ra các chiến lược kinh doanh phù hợp nhằm tăng cường sự hài lòng của khách hàng.

Đề tài *"Nghiên cứu và cài đặt chương trình phát hiện các tập mẫu phổ biến sử dụng thuật toán FP-Growth"* nhằm tìm hiểu và ứng dụng thuật toán FP-Growth trong khai phá tập mẫu phổ biến từ tập dữ liệu các giao dịch mua hàng. Thuật toán FP-Growth nổi bật nhờ khả năng hiệu quả trong việc nén dữ liệu và nhanh chóng phát hiện các tập mẫu thường xuất hiện, giúp tiết kiệm tài nguyên và thời gian tính toán so với các thuật toán trước đây như Apriori.

Trong khuôn khổ đề tài này, chúng tôi sử dụng tập dữ liệu từ các giao dịch mua hàng của khách hàng tại các cửa hàng tạp hóa, phân tích nhằm phát hiện các mặt hàng thường được mua cùng nhau. Kết quả từ quá trình phân tích sẽ giúp hình thành nên các nhóm sản phẩm tiềm năng, hỗ trợ các chiến lược kinh doanh như sắp xếp hàng hóa, khuyến mãi, hoặc phát triển sản phẩm theo nhu cầu thực tế của người tiêu dùng.

Chúng tôi hy vọng rằng kết quả từ nghiên cứu này sẽ góp phần cung cấp cái nhìn tổng quan về cách khai thác giá trị từ dữ liệu giao dịch và thể hiện tính hiệu quả của thuật toán FP-Growth trong khai phá dữ liệu thực tiễn.

**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

* 1. **Phát biểu đề tài**

Trong thời đại kỹ thuật số, lượng dữ liệu phát sinh từ các hoạt động kinh doanh ngày càng gia tăng, đặc biệt trong lĩnh vực bán lẻ và thương mại điện tử. Các doanh nghiệp hiện đại luôn nỗ lực để khai thác giá trị từ nguồn dữ liệu này nhằm tối ưu hóa dịch vụ, tăng cường trải nghiệm khách hàng và xây dựng chiến lược tiếp thị hiệu quả.

Đề tài *"Nghiên cứu và cài đặt chương trình phát hiện các tập mẫu phổ biến sử dụng thuật toán FP-Growth"* nhằm tập trung vào việc phát hiện các tập hợp sản phẩm thường được mua cùng nhau trong dữ liệu giao dịch của cửa hàng tạp hóa. Thông qua việc triển khai thuật toán FP-Growth, chương trình sẽ phát hiện ra các tập hợp sản phẩm phổ biến, từ đó giúp doanh nghiệp tối ưu hóa việc trưng bày sản phẩm, cải thiện các chiến lược khuyến mãi, và gia tăng mức độ hài lòng của khách hàng.

* 1. **Tính cấp thiết của đề tài**

Trong lĩnh vực bán lẻ, các nhà quản lý không chỉ quan tâm đến các sản phẩm riêng lẻ mà còn tìm hiểu các sản phẩm nào thường được mua cùng nhau. Điều này giúp tối ưu hóa trải nghiệm mua sắm của khách hàng, đồng thời mang lại cơ hội gia tăng doanh thu thông qua các chương trình khuyến mãi hoặc sắp xếp sản phẩm chiến lược. Ví dụ: Nếu hai sản phẩm A và B thường xuyên được mua cùng nhau, cửa hàng có thể bố trí chúng gần nhau hơn hoặc áp dụng các chương trình giảm giá khi mua đồng thời cả hai.

Trong các phương pháp khai phá tập mẫu phổ biến, thuật toán Apriori là phương pháp truyền thống, nhưng khi khối lượng dữ liệu tăng lên, Apriori dễ gặp phải các vấn đề về thời gian và bộ nhớ. Thuật toán FP-Growth, với cấu trúc FP-tree, được giới thiệu để giải quyết các hạn chế của Apriori bằng cách nén dữ liệu và giảm thiểu số lần quét cơ sở dữ liệu. Do đó, nghiên cứu về FP-Growth là cấp thiết và hữu ích, đặc biệt đối với các hệ thống phân tích dữ liệu lớn trong ngành bán lẻ.

* 1. **Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài tập trung vào:

* Sử dụng thuật toán FP-Growth để phát hiện các tập hợp sản phẩm phổ biến dựa trên tập dữ liệu các giao dịch tại cửa hàng tạp hóa.
* Khai thác bộ dữ liệu "Groceries\_dataset.csv", trong đó mỗi giao dịch chứa các sản phẩm mà khách hàng đã mua, và mỗi sản phẩm được phân tách bằng khoảng trắng trong cột itemDescription.
* Tập trung vào các thông số phổ biến trong khai phá tập mẫu như độ hỗ trợ (support) và độ tin cậy (confidence), không mở rộng đến các mô hình phức tạp hơn như mô hình dự đoán hoặc các phương pháp khai phá khác.

## **Đối tượng và mục đích nghiên cứu**

* **Đối tượng nghiên cứu**: Đối tượng của đề tài là tập dữ liệu giao dịch của khách hàng tại các cửa hàng tạp hóa, với mỗi giao dịch chứa danh sách các sản phẩm mà khách hàng đã mua.
* **Mục đích nghiên cứu**:
  + Xây dựng và triển khai một hệ thống sử dụng thuật toán FP-Growth để phát hiện các tập hợp sản phẩm phổ biến trong cơ sở dữ liệu giao dịch.
  + Phân tích các tập mẫu phổ biến được tìm thấy, đưa ra các gợi ý về cách cải thiện trải nghiệm khách hàng và các chiến lược kinh doanh dựa trên kết quả khai phá.
  1. Phương pháp nghiên cứu

Để đạt được mục tiêu đề tài, các phương pháp nghiên cứu được áp dụng như sau:

1. **Phân tích tài liệu:** Tìm hiểu lý thuyết về khai phá tập mẫu phổ biến và thuật toán FP-Growth từ các tài liệu, slide bài giảng và các nghiên cứu khoa học để hiểu rõ cách thức hoạt động của thuật toán và các ứng dụng thực tế.
2. **Thu thập và xử lý dữ liệu:** Sử dụng bộ dữ liệu "Groceries\_dataset.csv" làm cơ sở thực nghiệm. Tiền xử lý dữ liệu để đảm bảo dữ liệu ở dạng phù hợp cho thuật toán.
3. **Triển khai thuật toán FP-Growth:** Cài đặt thuật toán FP-Growth để phát hiện các tập mẫu phổ biến từ cơ sở dữ liệu. Bao gồm các bước xây dựng FP-tree, tạo cơ sở mẫu có điều kiện và tìm tập mẫu.
4. **Phân tích và đánh giá kết quả:** Đánh giá các tập mẫu phổ biến theo các tiêu chí đã đặt ra (ví dụ: ngưỡng độ hỗ trợ tối thiểu). Phân tích các mẫu phổ biến để đưa ra các khuyến nghị kinh doanh.
   1. Cấu trúc của đề tài

Cấu trúc đề tài bao gồm các chương sau đây:

* **Chương 1 - Giới thiệu đề tài:** Giới thiệu tổng quan về đề tài nghiên cứu, phát biểu đề tài, tính cấp thiết, phạm vi nghiên cứu, đối tượng và mục đích nghiên cứu. Phần này giúp người đọc hiểu được lý do lựa chọn đề tài và những giá trị thực tiễn mà đề tài mang lại.
* **Chương 2 - Cơ sở lý thuyết:** Trình bày các lý thuyết cơ bản liên quan đến khai phá dữ liệu, khai phá tập mẫu phổ biến, và thuật toán FP-Growth. Ngoài ra, chương này cũng bao gồm các ví dụ minh họa để làm rõ hơn về lý thuyết. Nếu có thể, sẽ đề cập đến một số cải tiến của thuật toán nhằm tăng hiệu quả khai phá và phân tích dữ liệu.
* **Chương 3 - Cài đặt thử nghiệm:** Cài đặt thử nghiệm chương trình phần mềm phát hiện các tập mẫu phổ biến trên bộ dữ liệu "Groceries\_dataset.csv". Chương này trình bày các bước triển khai thực tế của thuật toán FP-Growth, từ việc tiền xử lý dữ liệu đến quá trình xây dựng FP-tree và tìm các tập mẫu phổ biến. Ngoài ra, dữ liệu thử nghiệm cũng sẽ được trực quan hóa bằng các biểu đồ để người đọc có cái nhìn tổng quan và dễ dàng theo dõi hơn.
* **Chương 4 - Đánh giá kết quả và nghiên cứu:** Trình bày và phân tích các kết quả đạt được từ việc khai phá dữ liệu, đồng thời đề cập đến những mặt chưa đạt được trong quá trình nghiên cứu. Cuối cùng, chương này sẽ nêu ra các hướng phát triển tiềm năng trong tương lai, như áp dụng thuật toán FP-Growth trên các tập dữ liệu lớn hơn hoặc kết hợp với các kỹ thuật khai phá dữ liệu khác để cải thiện kết quả.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1. Khái niệm về khai phá dữ liệu**

***Khai phá dữ liệu*** *(Data Mining)* là một quá trình tìm kiếm các mẫu, xu hướng hoặc thông tin có ý nghĩa từ một lượng lớn dữ liệu. Mục tiêu của khai phá dữ liệu là biến đổi dữ liệu thô thành thông tin hữu ích, hỗ trợ cho việc ra quyết định trong các lĩnh vực như kinh doanh, y tế, khoa học và kỹ thuật.

Khai phá dữ liệu thường bao gồm các kỹ thuật như:

* ***Phân loại (Classification)****:* Gán các đối tượng dữ liệu vào các nhóm hoặc lớp đã biết trước.
* ***Hồi quy (Regression)****:* Dự đoán giá trị liên tục của dữ liệu.
* ***Phân cụm (Clustering):*** Nhóm các đối tượng dữ liệu tương tự nhau vào cùng một cụm.
* ***Khai phá tập mẫu phổ biến (Frequent Pattern Mining)****:* Tìm kiếm các tập hợp hoặc mẫu phổ biến trong cơ sở dữ liệu.

## **2.2. Khai phá tập mẫu phổ biến**

***Khai phá tập mẫu phổ biến*** là một kỹ thuật đặc biệt trong khai phá dữ liệu nhằm tìm ra các tập hợp hoặc mẫu xuất hiện thường xuyên trong cơ sở dữ liệu. Đối với bài toán khai phá dữ liệu bán lẻ, khai phá tập mẫu phổ biến cho phép phát hiện các sản phẩm thường được mua cùng nhau, giúp doanh nghiệp tối ưu hóa quy trình bán hàng và xây dựng các chiến lược tiếp thị hiệu quả.

Hai chỉ số chính trong khai phá tập mẫu phổ biến là:

* **Độ hỗ trợ (Support)**: Đo lường tần suất xuất hiện của một tập hợp sản phẩm trong tất cả các giao dịch. Độ hỗ trợ của tập hợp X được tính bằng công thức:

Ví dụ: Nếu sản phẩm {A, B} xuất hiện trong 3 giao dịch trong tổng số 10 giao dịch, độ hỗ trợ của {A, B} là 0.3.

* ***Độ tin cậy (Confidence)****:* Đo lường xác suất sản phẩm B xuất hiện trong một giao dịch khi sản phẩm A đã xuất hiện trong giao dịch đó. Độ tin cậy của luật kết hợp A → B được tính bằng công thức:

Ví dụ: Nếu có 5 giao dịch chứa {A, B} trong 7 giao dịch chứa {A}, thì độ tin cậy của luật {A} → {B} là 0.71.

## **2.3. Thuật toán FP-Growth**

#### *Thuật toán FP-Growth (Frequent Pattern Growth)* là một thuật toán quan trọng trong khai phá dữ liệu và phân tích dữ liệu. Nó được sử dụng để tìm kiếm các mẫu (patterns) thường xuyên trong một tập dữ liệu, được phát triển bởi Jiawei Han và các cộng sự vào năm 2000 và đã trở thành một trong những thuật toán phổ biến để khám phá luật kết hợp trong dữ liệu.

***Thuật toán FP-Growth (Frequent Pattern Growth)*** là một phương pháp phổ biến trong khai phá tập mẫu phổ biến. FP-Growth nổi bật nhờ khả năng tối ưu hóa thời gian và tài nguyên, đặc biệt phù hợp với các tập dữ liệu lớn.

##### 2.3.1. Nguyên lý của thuật toán FP-Growth

Thuật toán FP-Growth hoạt động theo hai bước chính:

* ***Xây dựng FP-tree****:* FP-tree là một cấu trúc cây nén, trong đó các nhánh của cây đại diện cho các tập hợp sản phẩm. Để xây dựng FP-tree, ta cần quét cơ sở dữ liệu một lần để tính tần suất của từng sản phẩm. Sau đó, cơ sở dữ liệu được quét lần thứ hai để xây dựng FP-tree theo thứ tự tần suất của sản phẩm, với sản phẩm có tần suất cao hơn được đặt trước.
* ***Khai phá FP-tree****:* Từ FP-tree, thuật toán tạo ra **cơ sở mẫu có điều kiện** cho từng sản phẩm, rồi từ đó xây dựng **FP-tree có điều kiện** cho từng sản phẩm này để tìm ra các tập hợp sản phẩm phổ biến.

##### 2.3.2. Ví dụ minh họa thuật toán FP-Growth

Giả sử ta có các giao dịch sau đây:

| **Giao dịch** | **Sản phẩm** |
| --- | --- |
| T1 | {A, B, C} |
| T2 | {B, C, D} |
| T3 | {A, C, D, E} |
| T4 | {A, B, C, D} |
| T5 | {A, B, D} |

Với ngưỡng độ hỗ trợ tối thiểu là 2, ta thực hiện các bước sau:

1. **Tính tần suất:** Đếm tần suất xuất hiện của từng sản phẩm:

* A: 4, B: 4, C: 3, D: 4, E: 1.

1. **Lọc sản phẩm theo ngưỡng hỗ trợ**: Sản phẩm E bị loại bỏ vì tần suất < 2.
2. **Sắp xếp sản phẩm theo tần suất**: Tạo lại các giao dịch với thứ tự sản phẩm giảm dần theo tần suất:

* T1: {A, B, C}
* T2: {B, C, D}
* T3: {A, C, D}
* T4: {A, B, C, D}
* T5: {A, B, D}

1. **Xây dựng FP-tree**: Sử dụng các giao dịch này để xây dựng FP-tree với các nhánh đại diện cho các giao dịch có thứ tự tần suất cao. Kết quả là một cây với các nhánh tương ứng, giúp phát hiện các tập mẫu phổ biến mà không cần quét lại toàn bộ cơ sở dữ liệu.
2. **Khai phá FP-tree**: Từ FP-tree, ta tạo cơ sở mẫu có điều kiện và tìm các tập hợp sản phẩm phổ biến.

## **2.4. Cải tiến thuật toán FP-Growth**

Một số cải tiến có thể giúp tối ưu thuật toán FP-Growth bao gồm:

* **Ngưỡng hỗ trợ động**: Cho phép thay đổi ngưỡng hỗ trợ theo từng giai đoạn hoặc theo từng tập con của dữ liệu, giúp thu hẹp không gian tìm kiếm và giảm thời gian khai phá.
* **Phân phối dữ liệu**: FP-Growth có thể được triển khai theo mô hình phân tán, sử dụng các công cụ như Hadoop hoặc Spark để xử lý dữ liệu lớn trong môi trường phân tán.
* **Nén dữ liệu bằng bảng băm**: Sử dụng bảng băm để giảm bớt kích thước của FP-tree, giúp giảm thiểu bộ nhớ khi xử lý các tập dữ liệu lớn.

## **2.5. Ưu và nhược điểm của thuật toán FP-Growth**

#### ****Ưu điểm****:

* + FP-Growth có khả năng nén dữ liệu tốt, giúp giảm số lần quét dữ liệu so với các thuật toán truyền thống như Apriori.
  + Cấu trúc FP-tree giúp tối ưu hóa thời gian và bộ nhớ.
  + Không cần phải tạo ra các tập hợp sản phẩm ứng viên, tiết kiệm thời gian tính toán.
* **Nhược điểm**:
  + FP-tree có thể tiêu tốn bộ nhớ lớn với các tập dữ liệu đa dạng.
  + Việc triển khai thuật toán FP-Growth yêu cầu kỹ thuật phức tạp, đặc biệt trong việc xây dựng và khai phá FP-tree.

## **2.6. Bài toán xây dựng cấu trúc FP-Tree**

***FP-Tree (Frequent Pattern tree)*** là cấu trúc dữ liệu của thuật toán FP-Growth để khai thác các tập mục phổ biến từ cơ sở dữ liệu bằng cách sử dụng các quy tắc kết hợp. Thuật toán được thiết kế để hoạt động dựa trên cơ sở dữ liệu chứa các giao dịch, chẳng hạn như lịch sử mua hàng của khách hàng trên trang web Amazon. Mặt hàng đã mua được coi là 'thường xuyên'. Tần suất tương tự sẽ chia sẻ nhánh tương tự của cây và khi chúng khác nhau, các nút sẽ phân chia chúng. Nút xác định một mục duy nhất từ nhánh (items) và nhánh (path) hiển thị số lần xuất hiện—các liên kết giữa các mục được gọi là liên kết nút (node)

**Bài toán**

Cho bảng dữ liệu bao gồm các giao dịch (tid) sau:

|  |  |
| --- | --- |
| TID | Items |
| 1 | f, a, c, d, g, i, m, p |
| 2 | a, b, c, f, l, m, o |
| 3 | b, f, h, j, o |
| 4 | b, c, k, s, p |
| 5 | a, f, c, e, l, p, m, n |

Tìm các tập mục có độ hỗ trợ ≥ 0.6 (tức tần số sup. ≥ 3).

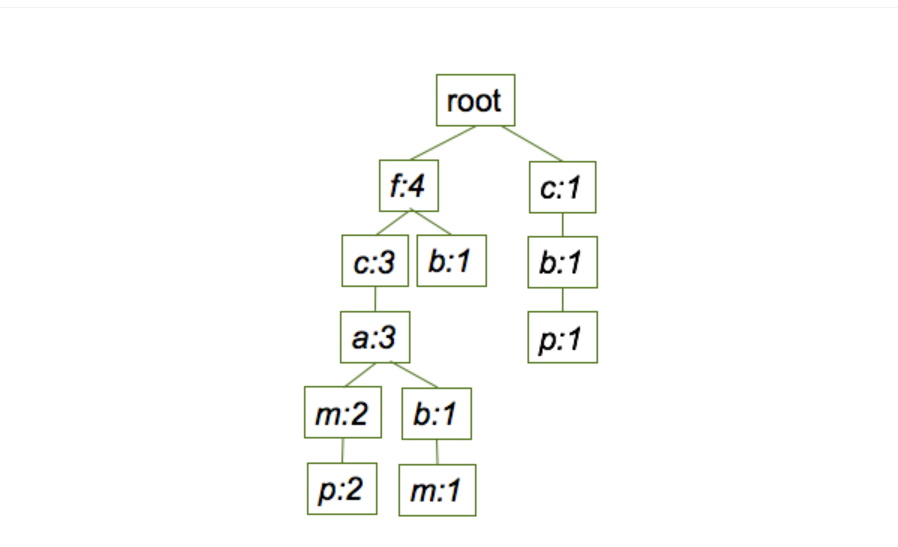
Đầu tiên tìm các item mức 1 có độ hỗ trợ ≥ 3, và sắp xếp theo thứ tự giảm dần:

|  |  |
| --- | --- |
| items | Items |
| f | 4 |
| c | 4 |
| a | 3 |
| b | 3 |
| m | 3 |
| p | 3 |

Tiếp theo sắp xếp các mục phổ biến mức 1 vừa tìm được theo thứ tự giảm dần trong mỗi giao dịch:

|  |  |
| --- | --- |
| items | Items |
| f, a, c, d, g, i, m, p | f, c, a, m, p |
| a, b, c, f, l, m, o | f, c, a, b, m |
| b, f, h, j, o | f, b |
| b, c, k, s, p | c, b, p |
| a, f, c, e, l, p, m, n | f, c, a, m, p |

Duyệt các Items phổ biến của mỗi giao dịch để xây dựng FP-Tree:



Tiếp theo, duyệt các item phổ biến mức 1 theo thứ tự tăng dần độ hỗ trợ là p, m, b, a, c, f. Với mỗi item, xây dựng các cơ sở mẫu điều kiện (conditional pattern-base) và sau đó là các FP-Tree điều kiện (conditional FP-Tree) của nó.

Tính chất: bất kì mẫu phổ biến nào có chứa mục Ii đều được chứa trên các nhánh (đường dẫn) của cây FP-Tree chứa Ii, số lần xuất hiện của mẫu chứa các nút trong đường dẫn tiền tố bằng số lần xuất hiện của nút Ii.

Bắt đầu với item p, cơ sở mẫu điều kiện của nó là tất cả các đường dẫn tiền tố của cây FP-Tree khi duyệt từ gốc root = null đến nút p, chính là {f,c,a,m}:2 và {c,b}:1 (số theo sau là số lần xuất hiện của nút p tương ứng với mỗi tiền tố đó).

Tiếp theo ta xây dựng FP-Tree điều kiện từ mẫu này bằng cách trộn tất cả các đường dẫn và giữ lại các nút có tổng các số đếm ≥ độ hỗ trợ = 3

{f,c,a,m}:2 và {c,b}:1 trộn lại thành f:2, c:3, a:2, m:2, b:1; chỉ có c:3 là thoả mãn điều kiện.

Do đó, các mẫu phổ biến chứa p là: p, {c,p}.

Làm tương tự cho các item còn lại, ta sẽ tìm được các mẫu phổ biến chứa các item đó. Cuối cùng có bảng sau đây:

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Cơ sở mẫu điều kiện | FP-Tree điều kiện | Các mẫu phổ biến |
| P | {f,c,a,m}:2, {c,b}:1 | c:3-p | p, {c,p} |
| m | {f,c,a}:2, {f,c,a,b}:1 | f:3,c:3, a:3-m | m,{f,m},{c,m},{a,m},{f,c,m},{c,a,m},{f,a,m}, {f,c,a,m} |
| b | {f,c,a}:1, f:1, c:1 | ∅ | b |
| a | {f,c}:3 | {f:3, c:3}-a | a, {f,a}, {c,a}, {f,c,a} |
| c | f:3 | {f:3}-c | c, {f,c} |
| f | ∅ | ∅ | f |

# CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM

## **3.1. Môi trường cài đặt**

* **Ngôn ngữ lập trình**:***Python*** là lựa chọn hàng đầu cho việc xử lý và phân tích dữ liệu vì tính dễ sử dụng và hỗ trợ từ cộng đồng lớn. Các thư viện liên quan đến khai phá dữ liệu cũng dễ dàng tích hợp vào Python, giúp tối ưu hóa quá trình triển khai thuật toán.
* **Các thư viện Python**:

**Pandas**: Đây là thư viện quan trọng nhất để quản lý dữ liệu trong Python. pandas cho phép chúng tôi dễ dàng tải dữ liệu từ file Groceries\_dataset.csv, xử lý, và tổ chức lại các thông tin cần thiết. Bên cạnh đó, pandas hỗ trợ các thao tác lọc, sắp xếp, và tóm tắt dữ liệu, giúp việc tiền xử lý dữ liệu trở nên thuận tiện.

**Numpy**: numpy hỗ trợ các phép toán trên mảng và ma trận, tối ưu hóa hiệu suất tính toán, đặc biệt hữu ích khi xử lý dữ liệu lớn và tính toán các chỉ số thống kê.

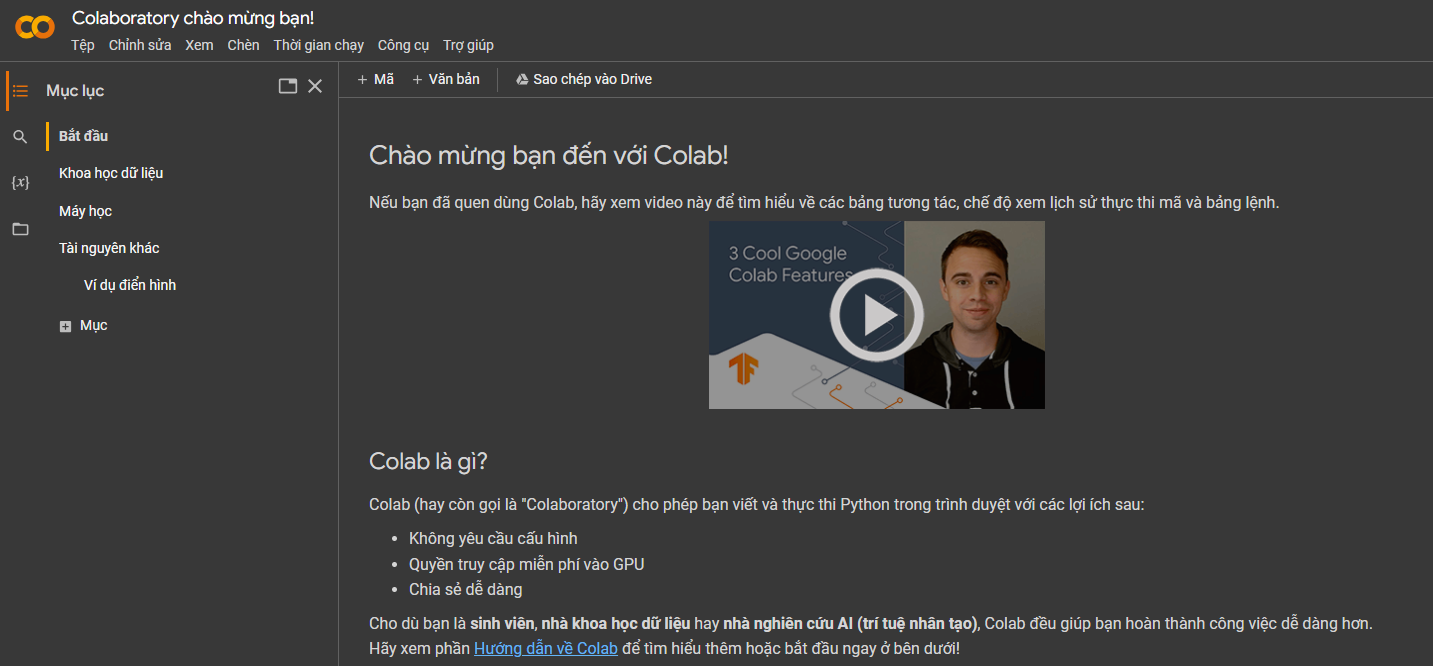
**Matplotlib** và **seaborn**: Các thư viện này dùng để tạo biểu đồ và hình ảnh minh họa kết quả. matplotlib cung cấp khả năng tạo biểu đồ cơ bản, trong khi seaborn giúp hiển thị dữ liệu theo dạng trực quan và dễ hiểu hơn. Điều này hỗ trợ phân tích và trình bày kết quả các tập mẫu phổ biến một cách rõ ràng.

**Graphviz**: Để hiển thị cấu trúc FP-tree, graphviz là công cụ mạnh mẽ giúp minh họa trực quan cây FP-tree, cho phép người dùng dễ dàng quan sát các nhánh của cây, tần suất của các mặt hàng và cấu trúc mẫu phổ biến.

* **Môi trường phát triển**:

***Colab*** hay còn gọi là ***"Colaboratory"*** cho phép bạn viết và thực thi Python trong trình duyệt với các lợi ích như không yêu cầu cấu hình, được quyền truy cập miễn phí vào GPU và chia sẻ dễ dàng

***Google Colab*** chỉ đơn giản là một đại diện trực tuyến của **Jupyter Notebook**. Mặc dù Jupyter Notebook cần cài đặt trên máy tính và chỉ có thể sử dụng tài nguyên máy cục bộ, nhưng Colab là một ứng dụng đám mây chính thức để mã hóa Python. Bạn có thể viết mã Python bằng Colab trên trình duyệt web **Google Chrome** hoặc **Mozilla Firefox** của mình. Cũng có thể thực thi các mã đó trên trình duyệt mà không cần bất kỳ môi trường thời gian chạy hoặc giao diện dòng lệnh nào.

****

Hình 3.1: Google colab

Dùng trong khoa học dữ liệu, bạn có thể khai thác toàn bộ sức mạnh của các thư viện Python phổ biến để phân tích và trực quan hóa dữ liệu. Ô chứa mã ở bên dưới sử dụng numpy để tạo một số dữ liệu ngẫu nhiên và sử dụng matplotlib để trực quan hóa dữ liệu đó.

Trong học máy, chỉ cần sử dụng một vài dòng mã là bạn có thể nhập tập dữ liệu hình ảnh, huấn luyện một trình phân loại hình ảnh dựa trên tập dữ liệu đó và đánh giá mô hình này. Sổ tay trên Colab thực thi mã trên các máy chủ đám mây của Google. Nhờ đó, bạn có thể tận dụng sức mạnh của phần cứng Google, bao gồm cả GPU và TPU, cho dù máy tính của bạn sử dụng sức mạnh phần cứng nào, bạn chỉ cần trình duyệt.

**3.2. Bộ dữ liệu**

**Tên bộ dữ liệu**: Groceries\_dataset.csv

1. **Mô tả bộ dữ liệu**:

* **Nguồn gốc**: Bộ dữ liệu giao dịch của các cửa hàng tạp hóa, mô phỏng các giao dịch mua hàng của khách hàng. Đây là một tập dữ liệu phổ biến trong các nghiên cứu khai phá tập mẫu phổ biến trong ngành bán lẻ.
* **Dạng dữ liệu**: Dữ liệu dạng văn bản, với mỗi dòng đại diện cho một giao dịch của khách hàng. Trong đó:
  + **itemDescription**: Cột này chứa danh sách các mặt hàng đã được mua trong một giao dịch. Các mặt hàng được phân tách nhau bằng dấu cách (khoảng trắng). Ví dụ: một giao dịch có thể bao gồm các mặt hàng như milk bread eggs.
  + Dữ liệu được tổ chức sao cho mỗi giao dịch là một đơn vị độc lập, điều này giúp thuận tiện cho việc triển khai FP-Growth để phân tích tần suất đồng xuất hiện của các mặt hàng trong các giao dịch.

1. **Đặc điểm của bộ dữ liệu**:

* **Kích thước và Phạm vi**: Bộ dữ liệu có hàng nghìn giao dịch, với mỗi giao dịch có thể chứa từ một đến nhiều mặt hàng. Phạm vi các mặt hàng bao gồm nhiều loại sản phẩm tạp hóa phổ biến, như thực phẩm (milk, bread), đồ uống (juice, soda), và đồ gia dụng.
* **Phân phối mặt hàng**: Do bộ dữ liệu này là tập hợp các giao dịch thực tế, số lượng xuất hiện của các mặt hàng sẽ có sự phân bổ không đồng đều, với một số mặt hàng phổ biến như milk, bread có tần suất xuất hiện cao hơn những mặt hàng ít phổ biến hơn.
* **Thách thức**: Số lượng mặt hàng lớn và cấu trúc không đồng nhất của các giao dịch khiến việc phân tích phải tập trung vào lọc và sắp xếp các mặt hàng theo tần suất để loại bỏ các mặt hàng ít xuất hiện.

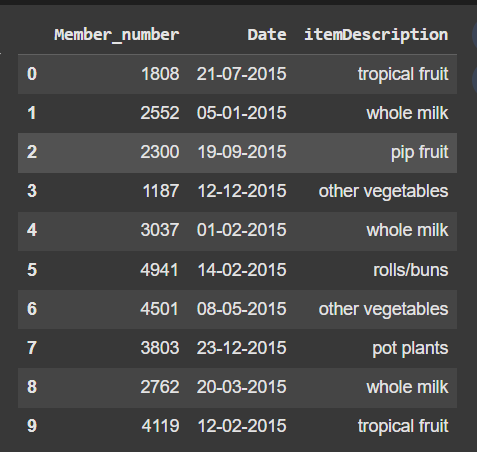
1. **Mục tiêu xử lý bộ dữ liệu**:

* **Xác định tập mẫu phổ biến**: Dùng thuật toán FP-Growth để tìm ra các tập hợp mặt hàng có xu hướng xuất hiện cùng nhau trong các giao dịch. Ví dụ, nếu milk và bread thường xuyên được mua cùng nhau, chúng sẽ trở thành tập mẫu phổ biến.
* **Ứng dụng thực tiễn**: Phân tích các tập mẫu phổ biến giúp tối ưu hóa các quyết định trong kinh doanh, ví dụ như cách xếp hàng hóa trong cửa hàng, đề xuất sản phẩm liên quan, và xây dựng các chương trình khuyến mãi cho các mặt hàng thường được mua cùng nhau.

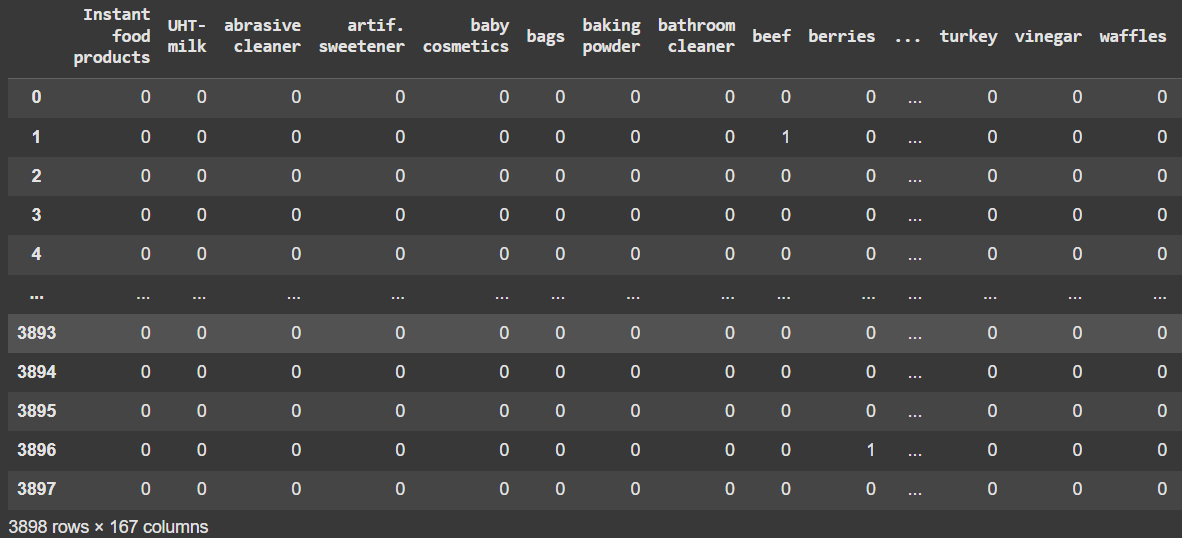
3.3. Quy trình phát triển ứng dụng AI

3.3.1 Thu thập và chuẩn bị bộ dữ liệu

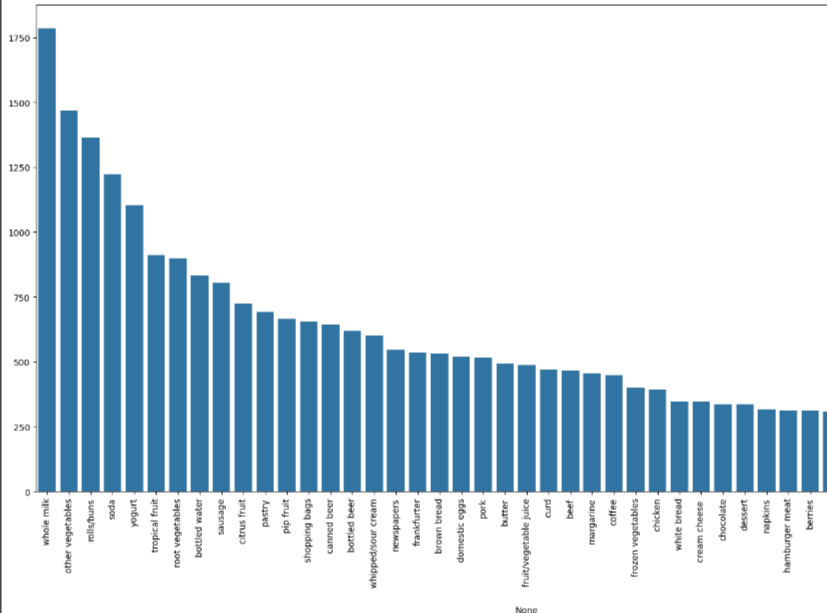
* Dữ liệu ban đầu



* Dữ liệu sau khi xử lý



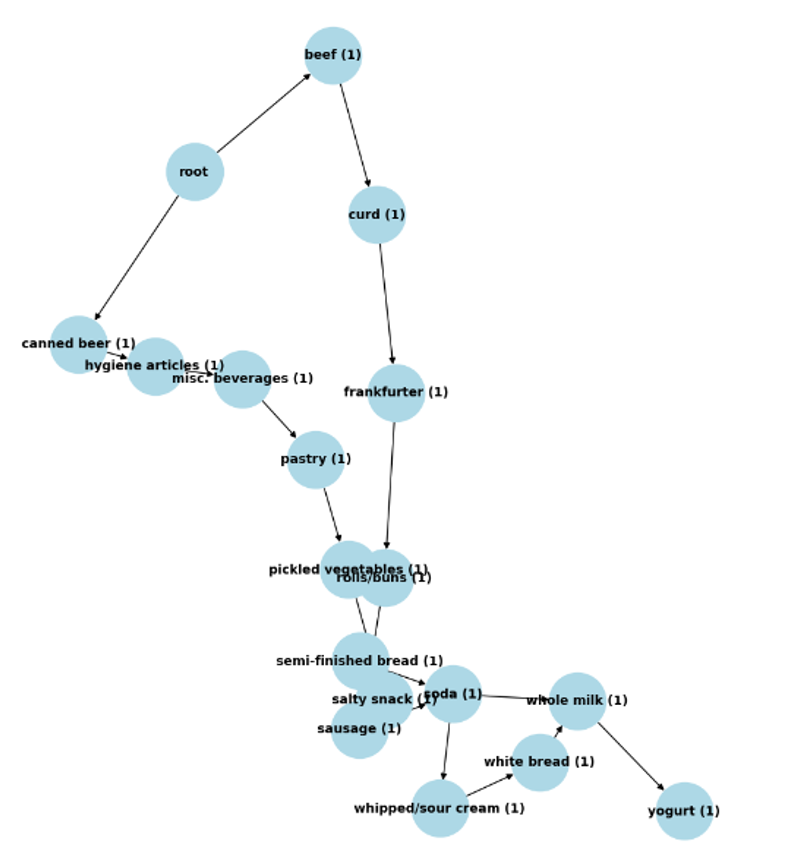
* Trực quan hoá dữ liệu
* Biểu đồ hình cột



* Biểu đồ heatmap



* Cây FP-tree



**CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ NGHIÊN CỨU**

## **4.1. Bảng so sánh ưu, nhược điểm của thuật toán Apriori và FP-Growth:**

|  |  |  |
| --- | --- | --- |
|  | Thuật toán Apriori | Thuật toán FP-Growth |
| Ưu điểm | |  |  | | --- | --- | | - Dễ hiểu và triển khai. |  |  |  |  |  | | --- | --- | --- | |  | - Thích hợp cho dữ liệu lớn. |  |  |  |  |  | | --- | --- | --- | |  | - Dễ dàng diễn giải các quy tắc kết hợp. |  | | - Tiết kiệm bộ nhớ với cấu trúc cây FP-tree.  - Chỉ cần quét cơ sở dữ liệu hai lần.  - Tốc độ nhanh hơn do không cần sinh tập ứng viên. |
| Nhược điểm | |  |  | | --- | --- | | - Tiêu tốn bộ nhớ khi số lượng tập ứng viên tăng. |  |  |  |  |  | | --- | --- | --- | |  | - Cần nhiều lần quét cơ sở dữ liệu, tăng thời gian xử lý. |  |  |  |  | | --- | --- | |  | - Hiệu suất giảm khi xử lý dữ liệu lớn. | | - Khó hiểu hơn với cấu trúc cây FP-tree.  - Hiệu quả nhất với cơ sở dữ liệu nhỏ và tập phổ biến không quá dài.  - Gặp khó khăn khi số lượng quy tắc lớn, tăng độ phức tạp trong xử lý. |

* 1. **Đánh giá kết quả nghiên cứu**

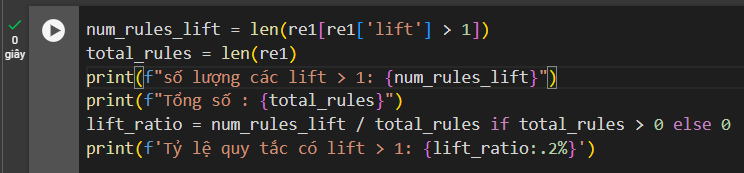
Đánh giá dựa trên các tiêu chí: Độ đo Lift và Conviction

**4.2.1. Độ đo lift:**

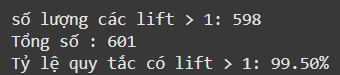
* Trong khai phá luật kết hợp, độ đo Lift (Lift measure) được sử dụng để đánh giá mức độ liên quan hoặc phụ thuộc giữa hai hoặc nhiều mặt hàng trong một tập dữ liệu giao dịch. Nói một cách đơn giản, Lift cho biết khả năng một mặt hàng được mua khi một mặt hàng khác đã được mua cao hơn bao nhiêu so với khi mặt hàng đó được mua một cách ngẫu nhiên.
* Xét luật kết hợp: A->B
* Độ đo lift(A -> B) được xác định như sau:

lift(A -> B) = conf(A -> B) / sup(B) = sup(A U B) / (sup(A).sup(B))

* Trong đó:
* sup(A U B) là tỷ lệ giao dịch chứa cả mặt hàng A và mặt hàng B.
* sup(A) là tỷ lệ giao dịch chứa mặt hàng A.
* sup (B) là tỷ lệ giao dịch chứa mặt hàng B.
* Nếu lift(A -> B) = 1: A và B độc lập, không nến có mối quan hệ tương quan giữa A và B
* Nếu lift(A -> B) > 1: luật A -> B có ý nghĩa (tương quan dương – positive correlation)
* Nếu lift(A -> B) < 1: luật A -> B và luật B -> A không có ý nghĩa (tương quan âm – negative correlation)



Sau khi đánh giá mô hình ta có kết quả sau:

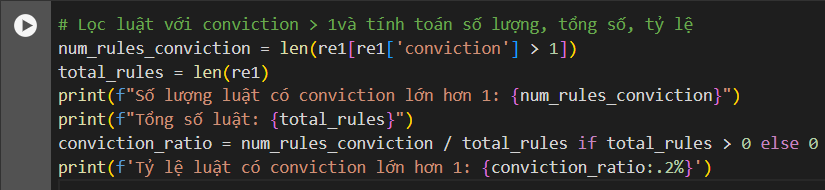


**4.2.2. Độ đo conviction:**

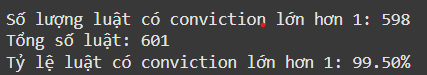
* Trong khai phá luật kết hợp, độ đo Conviction được sử dụng để đánh giá mức độ ngụ ý của một luật. Nó đo lường mức độ mà hậu quả của một luật (consequent) phụ thuộc vào tiền đề của nó (antecedent). Nói cách khác, Conviction cho biết sự tin tưởng vào một luật sẽ sai lệch như thế nào nếu tiền đề và hậu quả là độc lập với nhau.
* Công thức tính Conviction:

Conviction(A -> B) = (1 - Support(B)) / (1 - Confidence(A -> B))

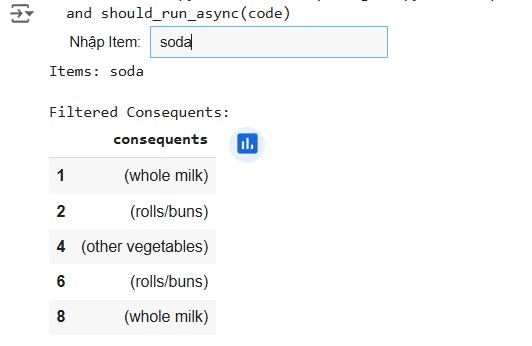
* Trong đó:
* Support(B) là tỷ lệ giao dịch chứa hậu quả B.
* Confidence(A -> B) là tỷ lệ giao dịch chứa cả tiền đề A và hậu quả B trong số các giao dịch chứa tiền đề A.
* Conviction = 1: Tiền đề và hậu quả là độc lập với nhau, luật không có ý nghĩa.
* Conviction > 1: Tiền đề và hậu quả có liên quan với nhau, luật có ý nghĩa. Giá trị Conviction càng cao, mức độ ngụ ý của luật càng mạnh.
* Dựa trên bộ dữ liệu thực tế có đánh giá sau:



Sau khi đánh giá mô hình ta có kết quả sau:



* 1. Xây dựng và tích hợp ứng dụng AI trên model



1. **Ứng dụng thực tiễn**

Thuật toán FP-Growth (Frequent Pattern Growth) là một kỹ thuật rất hiệu quả trong khai thác mẫu thường xuyên từ tập dữ liệu lớn. Dưới đây là một số ứng dụng thực tế của thuật toán FP-Growth:

Phân tích giỏ hàng: FP-Growth thường được sử dụng trong lĩnh vực bán lẻ để phát hiện các sản phẩm thường xuyên được mua cùng nhau. Điều này giúp các nhà bán lẻ xác định các combo sản phẩm, tối ưu hóa trưng bày sản phẩm và tăng doanh thu.

Đề xuất sản phẩm: Các hệ thống gợi ý (recommendation systems) có thể sử dụng FP-Growth để đưa ra các gợi ý dựa trên các mẫu giao dịch của người tiêu dùng. Ví dụ, nếu một khách hàng mua một sản phẩm, hệ thống có thể gợi ý các sản phẩm khác thường được mua cùng.

Phân tích mạng xã hội: Trong các mạng xã hội, FP-Growth có thể được sử dụng để tìm ra các mẫu tương tác hoặc kết nối thường xuyên giữa người dùng, từ đó phát hiện ra các nhóm hoặc cộng đồng.

Phân tích hành vi người dùng: Các công ty công nghệ có thể áp dụng FP-Growth để phân tích hành vi của người dùng trên các nền tảng trực tuyến, từ đó tối ưu hóa trải nghiệm người dùng và cá nhân hóa nội dung.

Khai thác dữ liệu y tế: Trong lĩnh vực y tế, FP-Growth có thể được sử dụng để phát hiện các mẫu bệnh lý thường gặp trong hồ sơ bệnh án, từ đó hỗ trợ bác sĩ trong việc chẩn đoán và điều trị.

Phân tích văn bản: FP-Growth có thể được áp dụng để phát hiện các mẫu từ vựng thường gặp trong một tập hợp tài liệu, giúp cải thiện hệ thống tìm kiếm và phân loại tài liệu.

Phát hiện gian lận: FP-Growth có thể giúp phát hiện các giao dịch gian lận trong lĩnh vực tài chính bằng cách tìm ra các mẫu giao dịch không bình thường.

FP-Growth là một thuật toán rất mạnh mẽ và linh hoạt, có thể áp dụng cho nhiều lĩnh vực khác nhau, từ thương mại điện tử đến nghiên cứu khoa học.

# CÁC TÀI LIỆU THAM KHẢO

[1] Slide bài giảng Nhập môn khai phá dữ liệu và học máy (Introduction to data mining and machine learning) của TS.Lưu Minh Tuấn trường Đại học

[2] <https://www.jaist.ac.jp/~bao/VNAlectures/AssociationAnalysis-Hieu.pdf>

[3] Giáo trình Học máy của thầy Hoàng Xuân Huấn Đại học Quốc Gia Hà Nội

<https://res.cloudinary.com/elearning/image/upload/v1465532195/575a3705104df31100a08ed6-1465532195268.pdf>

[4] Bài giảng Học máy ngành Khoa học Máy tính của PGS.TS. Nguyễn Quang Hoan trường Đại học Thuỷ Lợi

[5]https://fr.slideshare.net/slideshow/data-mining-lut-kt-hp-v-ng-dng/46902428