**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN MÔN HỌC**

**KHAI PHÁ DỮ LIỆU**

**Đề tài:**

**Khai phá tập dữ liệu Tiếp thị ngân hàng (Bank Marketing)**

**Nhóm sinh viên thực hiện : ĐÀO MINH KHOA**

**NGUYỄN QUỐC HIỆU**

**TÔ VĂN TUẤN**

**Giảng viên phụ trách môn học : TRẦN MẠNH TUẤN**

**Ngành : CÔNG NGHỆ THÔNG TIN**

**Khóa : 59**

*Hà Nội, tháng 05 năm 2021*

**LỜI MỞ ĐẦU**

Trong thời buổi hiện đại ngày nay, công nghệ thông tin cũng như những ứng dụng của nó không ngừng phát triển, lượng thông tin và cơ sở dữ liệu được thu thập và lưu trữ cũng tích lũy ngày một nhiều lên Con người cũng vì thế mà cần có thông tin với tốc độ nhanh nhất để đưa ra quyết định dựa trên lượng dữ liệu khổng lồ đã có. Các phương pháp quản trị và khai thác cơ sở dữ liệu truyền thống ngày càng không đáp ứng được thực tế, vì thế, một khuynh hướng kỹ thuật mới là Kỹ thuật phát hiện tri thức và khai phá dữ liệu nhanh chóng được phát triển.

Khai phá dữ liệu đã và đang được nghiên cứu, ứng dụng trong nhiều lĩnh vực khác nhau ở các nước trên thế giới. Ở Việt Nam, kỹ thuật này đang được nghiên cứu và dần đưa vào ứng dụng. Khai phá dữ liệu là một bước trong quy trình phát hiện tri thức. Hiện nay, mọi người không ngừng tìm tòi các kỹ thuật để thực hiện khai phá dữ liệu một cách nhanh nhất và có được kết quả tốt nhất.

Trong bài tập lớn này, chúng em tìm hiểu và sử dụng hai kỹ thuật đó là: phân lớp và khai phá luật kết hợp với đề tài “Khai phá dữ liệu trong lĩnh vực tiếp thị ngân hàng” trên tập dữ liệu Bank Marketing.

Mục lục

[Chương 1. Tổng quan về Khai phá dữ liệu 5](#_Toc43982504)

[1.1. Khái niệm 5](#_Toc43982505)

[1.2. Ứng dụng 6](#_Toc43982506)

[1.3. Phần mềm khai phá 7](#_Toc43982507)

[1.4. Mô tả dữ liệu 7](#_Toc43982508)

[Chương 2. Tiền xử lý dữ liệu 9](#_Toc43982509)

[2.1. Dữ liệu thô 9](#_Toc43982510)

[2.2. Tiền xử lý 9](#_Toc43982511)

[2.2.1. Chuyển đổi định dạng file 9](#_Toc43982512)

[2.2.2. Quan sát dữ liệu 12](#_Toc43982513)

[2.2.3. Làm sạch dữ liệu 14](#_Toc43982514)

[3.1.1. Rút gọn dữ liệu 17](#_Toc43982515)

[Chương 3. Khai phá dữ liệu 18](#_Toc43982516)

[3.1. Phân lớp dữ liệu 18](#_Toc43982517)

[3.1.1. Khái niệm 18](#_Toc43982518)

[3.1.2. Ứng dụng 20](#_Toc43982519)

[3.1.3. Mô hình phân lớp 20](#_Toc43982520)

[3.1.4. Các tiêu chuẩn của mô hình phân lớp 20](#_Toc43982521)

[3.1.5. Cây quyết định ứng dụng trong phân lớp 21](#_Toc43982522)

[3.1.6. Ưu nhược điểm của cây quyết định 23](#_Toc43982523)

[3.1.7. Thuật toán C4.5 23](#_Toc43982524)

[3.1.8. Sử dụng thuật toán C4.5 để phân lớp trong Weka 27](#_Toc43982525)

[3.2. Luật kết hợp 40](#_Toc43982526)

[3.2.1. Cơ sở lý thuyết 40](#_Toc43982527)

[3.2.2. Định nghĩa 40](#_Toc43982528)

[3.2.3. Các bước thực hiện khai phá luật kết hợp và ý nghĩa thông số 48](#_Toc43982529)

[Chương 4. Biểu diễn dữ liệu 62](#_Toc43982530)

[4.1. Phân lớp dữ liệu 62](#_Toc43982531)

[4.2. Luật kết hợp 65](#_Toc43982532)

[Chương 5. Nhận xét đánh giá 69](#_Toc43982533)

[5.1. Nhận xét đánh giá 69](#_Toc43982534)

[5.2. Kết luận 69](#_Toc43982535)

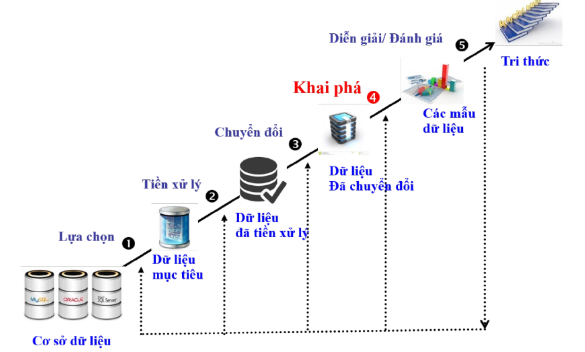
[5.3. Tài liệu tham khảo 69](#_Toc43982536)

**DANH MỤC CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Nghĩa | Giải thích |
| CSDL | Cơ sở dữ liệu | Tập hợp các dữ liệu có cấu trúc và liên quan với nhau được lưu trữ trên máy tính |
| WEKA | Waikato Enviroment for Knowledge Anylysis | Phần mềm dùng để khai phá dữ liệu |

1. **Tổng quan về Khai phá dữ liệu**
   1. **Khái niệm**

* Khai phá dữ liệu là một quá trình xác định các mẫu tiềm ẩn có tính hợp lệ, mới lạ, có ích và có thể hiểu được trong một khối dữ liệu rất lớn.
* Khai phá tri thức từ CSDL (Knowledge Discovery in DataBase) gồm 5 bước:
  + Bước 1: Lựa chọn CSDL
  + Bước 2: Tiền xử lý
  + Bước 3: Chuyển đổi
  + Bước 4: Khai phá dữ liệu
  + Bước 5: Diễn giải và đánh giá.
* Khai phá dữ liệu là 1 bước trong quá trình khai phá tri thức từ CSDL



Hình 1-1: Quy trình khai phá tri thức

* 1. **Ứng dụng**

Phát hiện tri thức và khai phá dữ liệu liên quan đến nhiều ngành, nhiều lĩnh vực: Thống kê, trí tuệ nhân tạo, CSDL, thuật toán, tính toán song song và tốc độ cao, thu thập tri thức cho các hệ chuyên gia, quan sát dữ liệu,…Đặc biệt phát hiện tri thức và khai phá dữ liệu gần gữi với lĩnh vực thống kê, sử dụng các phương pháp thồng kê để mô hình dữ liệu và phát hiện các mẫu, luật… Ngân hàng dữ liệu (Data Warehousing) và các công cụ phân tích trực tuyến cũng liên quan rất chặt chẽ với phát hiện tri thức và khai phá dữ liệu.

* Khai phá dữ liệu có nhiều ứng dụng trong thực tế, ví dụ như:
  + Bảo hiểm, tài chính và thị trường chứng khoán phân tích tình hình tài chính và dự báo giá của các loại cổ phiếu trong thị trường chứng khoán. Danh mục vốn và giá, lãi suất, dữ liệu thẻ tín dụng, phát hiện gian lận, ...
  + Thống kê, phân tích dữ liệu và hỗ trợ ra quyết định.
  + Điều trị y học và chăm sóc y tế: một số thông tin về chuẩn đoán bệnh lưu trong các hệ thống quản lý bệnh viện Phân tích mối liên hệ giữa các triệu chứng bệnh, chẩn đoán và phương pháp điều trị (chế độ dinh dưỡng,thuốc, ...)
  + Sản xuất và chế biến: Quy trình, phương pháp chế biến và xử lý sự cố.
  + Text mining và Web mining: Phân lớp văn bản và các trang Web, tóm tắt văn bản,... đi
  + Lĩnh vực khoa học: Quan sát thiên văn dữ liệu gene, dữ liệu sinh vật học, tìm kiếm, so sánh các hệ gen và thông tin di truyền, mối liên hệ gen và một số bệnh di truyền, .. .
  + Mạng viễn thông: Phân tích các cuộc gọi điện thoại và hệ thống giám sát lỗi, sự cố, chất lượng dịch vụ ...
  1. **Phần mềm khai phá**
* WEKA (Waikato Enviroment for Knowledge Anylysis) là một tập hợp các giải thuật học máy và các công cụ xử lý dữ liệu.
* Ngôn ngữ: Java
  1. **Mô tả dữ liệu**

Thông tin tập dữ liệu:

* Dữ liệu liên quan đến các chiến dịch tiếp thị trực tiếp của một tổ chức ngân hàng tại Bồ Đào Nha. Các chiến dịch tiếp thị được dựa trên các cuộc gọi điện thoại.
* Mục tiêu của việc phân loại là để dự đoán khách hàng sẽ đăng ký một khoản tiền gửi có kỳ hạn hay không.

Thông tin thuộc tính:

* Biến đầu vào (Dữ liệu khách hàng của ngân hàng):
  + Age: Độ tuổi
  + Job: Công việc (Bao gồm: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired','self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
  + Marital: Tình trạng hôn nhân (Bao gồm: 'divorced', 'married', 'single', 'unknown')
  + Education: Cấp bậc giáo dục (Bao gồm: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')
  + Default: Trả nợ đúng kỳ hạn hay không? (Bao gồm: 'no', 'yes', 'unknown')
  + Housing: Có khoản vay nào về nhà ở hay không? (Bao gồm: 'no', 'yes', 'unknown')
  + Loan: Có khoản vay cá nhân hay không? (Bao gồm: 'no', 'yes', 'unknown')
  + Contact: Loại liên hệ (Bao gồm: 'cellular', 'telephone')
  + Month: Tháng liên hệ cuối cùng của năm (Bao gồm: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
  + Day\_of\_week: Ngày liên hệ cuối cùng trong tuần (Bao gồm: 'mon', 'tue', 'wed', 'thu', 'fri')
  + Duration: Thời lượng liên lạc cuối cùng (tính bằng giây).

Lưu ý quan trọng: thuộc tính này ảnh hưởng lớn đến mục tiêu đầu ra (ví dụ: nếu thời gian = 0 thì y = 'không'). Tuy nhiên, thời lượng không được biết trước khi cuộc gọi được thực hiện. Ngoài ra, sau khi kết thúc cuộc gọi y rõ ràng là đã biết. Như vậy đầu vào này chỉ nên được đưa vào cho mục đích điểm chuẩn và nên được loại bỏ nếu ý định muốn có một mô hình dự đoán thực tế.

* + Campaign: Số lượng liên hệ được thực hiện trong chiến dịch này và cho khách hàng này
  + Pdays: số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước đó (999 có nghĩa là khách hàng đã không liên hệ trước đó).
  + Previous: số lượng liên hệ được thực hiện trước chiến dịch này và cho khách hàng này.
  + Poutcome: kết quả của chiến dịch tiếp thị trước đó (Bao gồm: 'failure', 'nonexistent', 'success')
  + Emp.var.rate: tỷ lệ biến đổi việc làm - chỉ số hàng quý
  + Cons.price.idx: chỉ số giá tiêu dùng - chỉ số hàng tháng
  + Cons.conf.idx: chỉ số niềm tin của người tiêu dùng - chỉ số hàng tháng
  + Euribor3m: Tỷ lệ chào bán liên ngân hàng Euribor trong 3 tháng - chỉ số hàng ngày
  + Nr.employed: số lượng nhân viên - chỉ số hàng quý
* Biến đầu ra (mong muốn mục tiêu):
  + Y - khách hàng đã đăng ký tiền gửi có kỳ hạn chưa? (Bao gồm: 'có', 'không')

1. **Tiền xử lý dữ liệu**
   1. **Dữ liệu thô**

Dữ liệu ban đầu có 41.189 bản ghi ở dạng csv. Tuy nhiên, dữ liệu không được chia theo cột mà lại cách nhau bởi dấu “;”

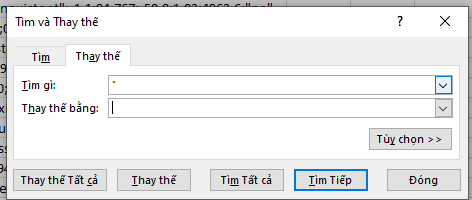


Hình 2-1: File dữ liệu csv gốc

* 1. **Tiền xử lý** 
     1. **Chuyển đổi định dạng file**

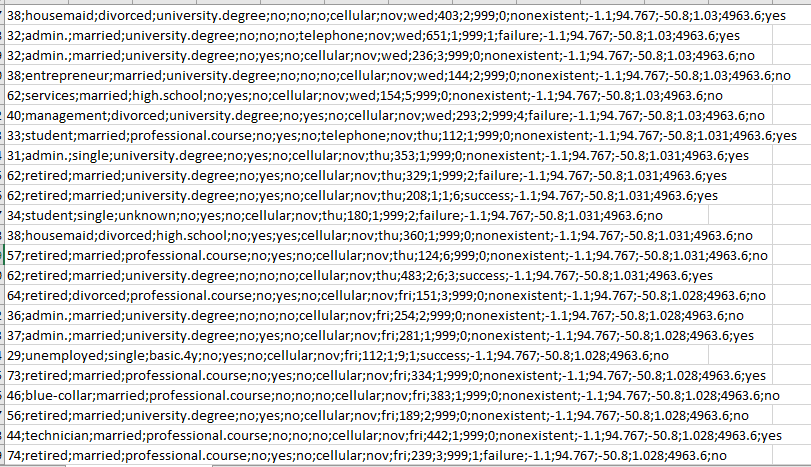
Ta tiến hành chuyển dữ liệu từ tập csv trên bằng Excel.

Bước 1: Tiến hành bỏ tất cả dấu “” bằng cách nhấn tổ hợp phím Ctrl + A sau đó nhấn tiếp Ctrl + F



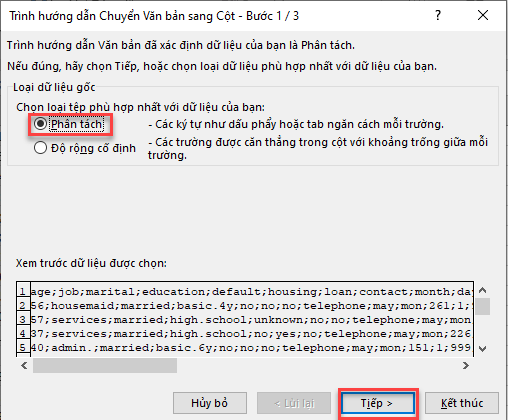
Hình 2-2: Tiến hành thay thế đồng loạt

Dữ liệu sau khi thay thế:



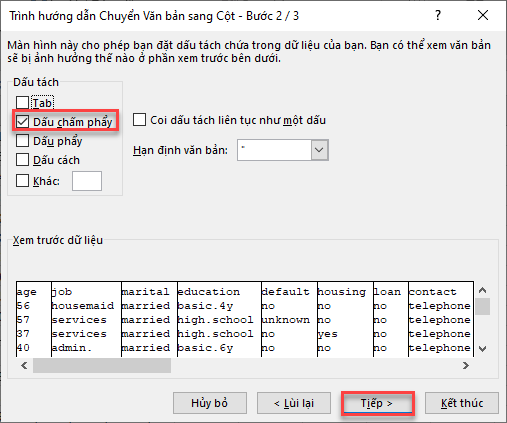
Hình 2-3: File dữ liệu csv sau khi thay thế

Bước 2: Chuyển đổ định dạng từ văn bản cách nhau bằng dấu “;” sang định dạng văn bản hàng cột bằng cách chọn Data (Dữ liệu) trên thanh công cụ và chọn Text to Column (Văn bản thành cột).



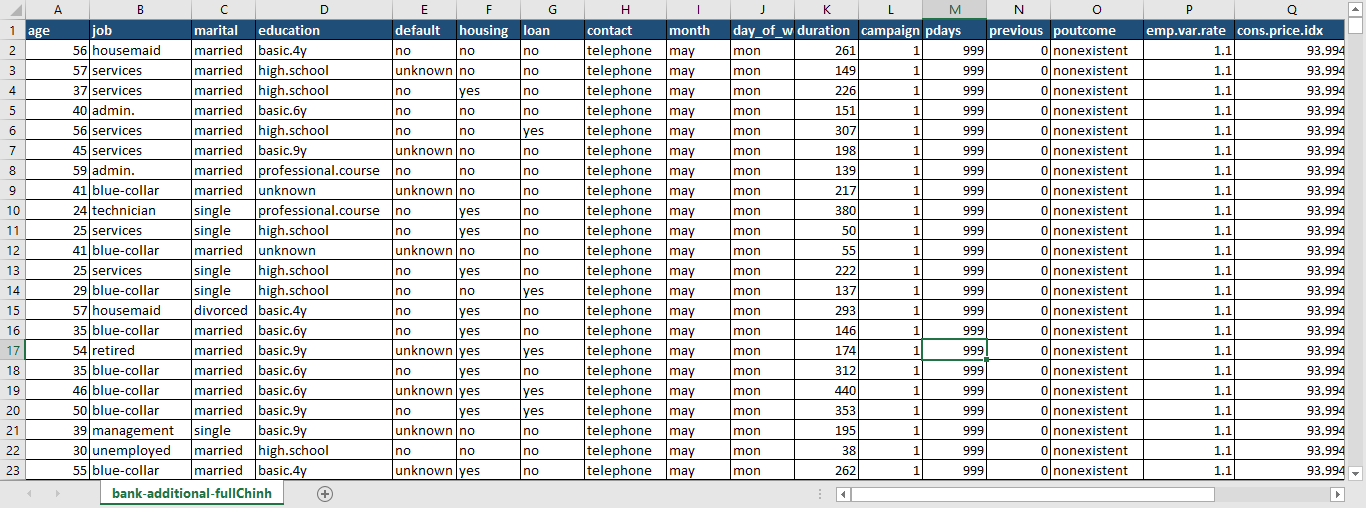
Hình 2-4: Phân tách dữ liệu từ văn bản thành cột

Bước 3: Chọn phương pháp tách cách nhau bởi “Dấu chấm phẩy”.



Hình 2-5: Phân tách dữ liệu từ văn bản thành cột cách nhau bởi dấu phẩy

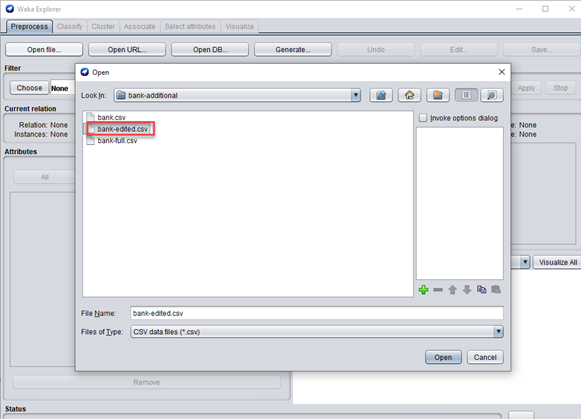
Bước 4: Chọn “Kết thúc” để hoàn tất quá trình chuyển đổi. Kết quả thu được đã chuyển sang dạng cột và Weka có thể đọc được tệp dữ liệu.



Hình 2-6: Dữ liệu sau khi phân tách thành cột

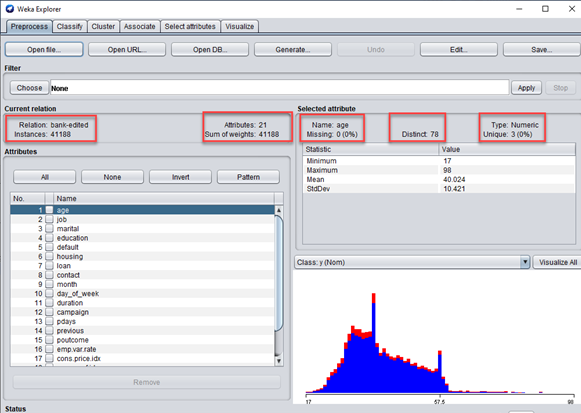
* + 1. **Quan sát dữ liệu**

Đầu tiên, ta dùng Weka để nạp dữ liệu bank-edited.csv vào chương trình như hình dưới. Ở tab Preprocessing, ta click vào nút Open files. Sau đó, tìm đến nơi lưu file bank-edited.csv và click Open (lưu ý: Files of Type ta chọn CSV data file (\*.csv)).



Hình 2-7: Đưa file đã chuyển định dạng vào Weka

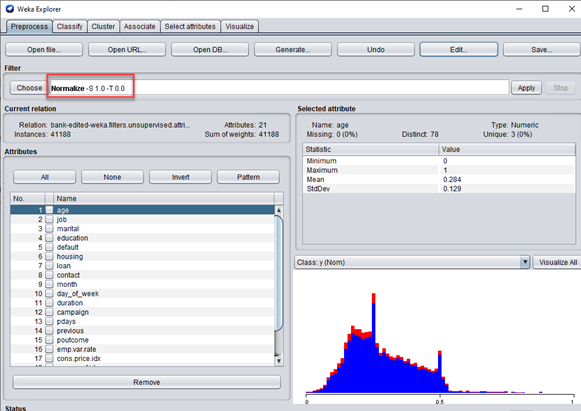
Trong tập dữ liệu tải về, ta quan sát thấy có 41.188 mẫu dữ liệu, 21 thuộc tính, và 0% dữ liệu bị thiếu giá trị. Tuy nhiên, kiểu dữ liệu đang ở dạng Numeric, nên chúng ta cần chuyển đổi về dạng Nominal để có thể thực hiện khai phá luật kết hợp.



Hình 2-8: Màn hình hiển thị dữ liệu từ file đầu vào

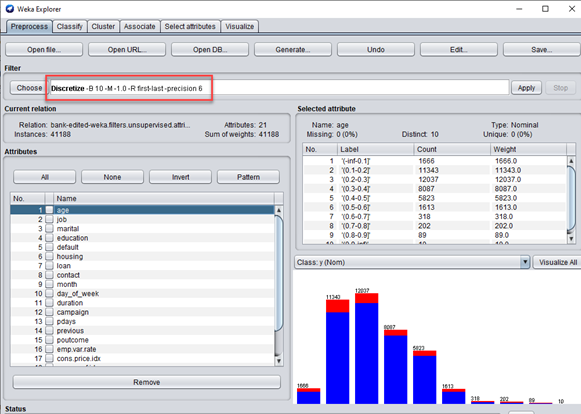
* + 1. **Làm sạch dữ liệu**

Trước tiên, ta chuẩn hóa các thuộc tính số về đoạn [0, 1] bằng bộ lọc Normalize.



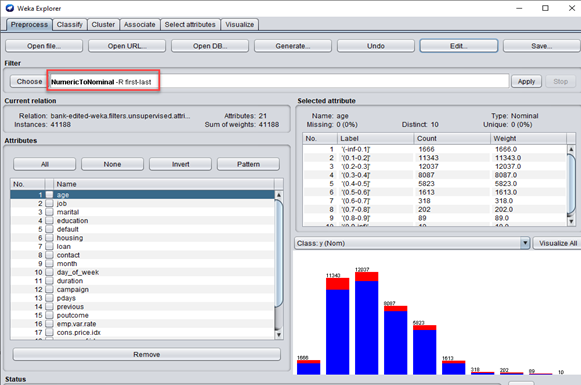
Hình 2-9: Làm sạch dữ liệu bằng bộ lọc Normalize

Tiếp đến, ta dùng bộ lọc Discretize: là bộ lọc dùng để rời rạc hóa các thuộc tính numeric thành nomial. Việc rời rạc đơn giản bằng cách chia giỏ (binning), sắp xếp và chia dữ liệu vào các giỏ có cùng độ rộng (equal-width). Chia vùng giá trị thành N khoảng cùng kích thước, Độ rộng của từng khoảng = (giá trị lớn nhất – giá trị nhỏ nhất)/N. Mặc định, Weka gán N=10.



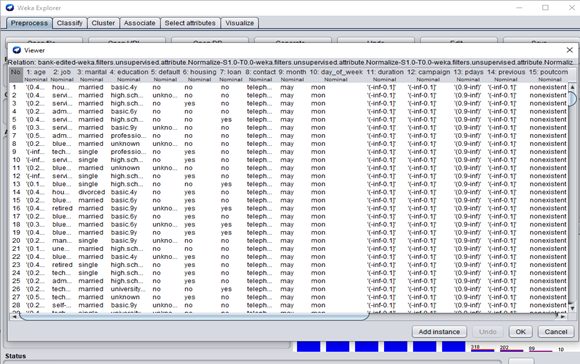
Hình 2-10: Rời rạc hóa thuộc tính bằng bộ lọc Discretize

Cuối cùng, ta dùng bộ lọc NumericToNominal: là bộ lọc dùng để chuyển các thuộc tính dạng numeric thành nomial. Không như discretization (rời rạc hóa), bộ lọc này gom các mẫu có cùng giá trị vào cùng một nhóm nomial.



Hình 2-11: Chuyển kiểu dữ liệu từ Numeric sang Nominal

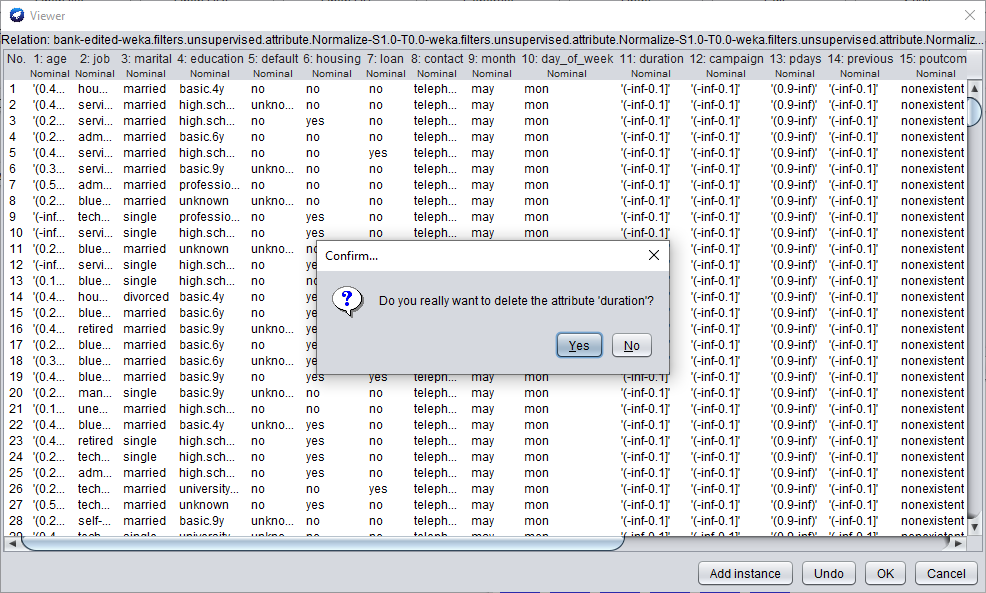
Kết quả thu được tất cả ở dạng Nominal



Hình 2-12: Chuyển kiểu dữ liệu từ Numeric sang Nominal

* + 1. **Rút gọn dữ liệu**

Như đã mô tả ở mục 1.4 Mô tả dữ liệu thì thuộc tính “Duration” nên được loại bỏ nếu muốn có một mô hình dự đoán thực tế. Vì vậy, chúng ta thực hiện xóa thuộc tính này trên weka bằng cách vào Edit -> Chuột phải và chọn Delete attribute.



Hình 2-13: Loại bỏ thuộc tính Duration

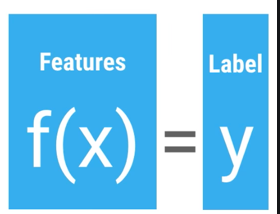
1. **Khai phá dữ liệu**
   1. **Phân lớp dữ liệu**
      1. **Khái niệm**

Phân lớp dữ liệu (Data Classification) là xếp đối tượng dữ liệu vào một trong các lớp đã đước xác định trước.Vì Cơ sở dữ liệu thường chứa các thông tin ẩn – các thông tin có thể phục vụ cho quá trình phân lớp nên các giải thuật phân lớp thường phân tích dữ liệu nhằm tìm ra các mô hình mô tả các lớp dữ liệu, từ đó có thể quyết định được một phần tử mới là thuộc vào lớp nào.

Phân lớp là một trong những mối quan tâm nhiểu nhất của con người trong quá trình làm việc với một tập hợp đối tượng.Việc này giúp con người có thể tiến hành sắp xếp và tìm kiếm các đối tượng một cách nhanh chóng.Hỗ trợ quá trình ra quyết định thông minh thậm chí là những quyết định mang tinh sống còn ví dụ một nhân viên vay vốn rất muốn biết khách hàng thuộc lớp “an toàn ” hay “mạo hiểm ” khi quyết định một đơn vay vốn mới .

Bản chất bài toán phân lớp là dự đoán các nhãn (hay lớp ) của các phần tử dữ liệu đầu vào hay các nhãn (hay lớp ) này là các giá trị rời rạc.

Giải thuật phân lớp ra hàm y = f(x) để từ đó khi có một phần tử x mới nó sẽ dự đoán nhãn y tương ứng với nó.



Hình 3-1: Giải thuật phân lớp

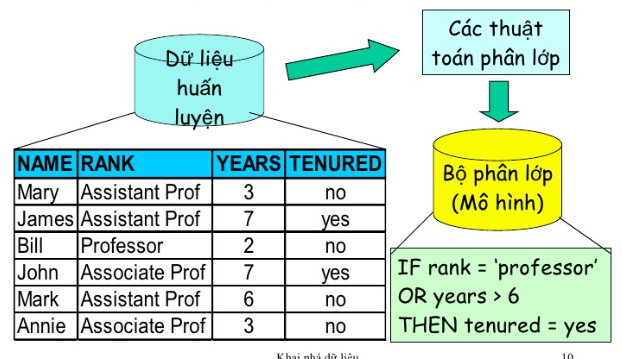
Quá trình phân lớp dũ liệu gồm 2 bước:

Bước 1:Xây dựng mô hình

Mô tả tập các lớp xác định.

Bước học (learning step) hay pha học (learning phase) là phân tích dữ liệu đã gán nhãn tìm ra mô hình phù hợp mô tả tập dữ liệu đó.Tập dữ liệu gán nhãn phục vụ quá trình học này được gọi là dữ liệu huấn luyện (training data).

Tìm luật phân lớp, cây quyết định hoặc công thức toán mô tả lớp.



Hình 3-2: Mô hình phân lớp 1

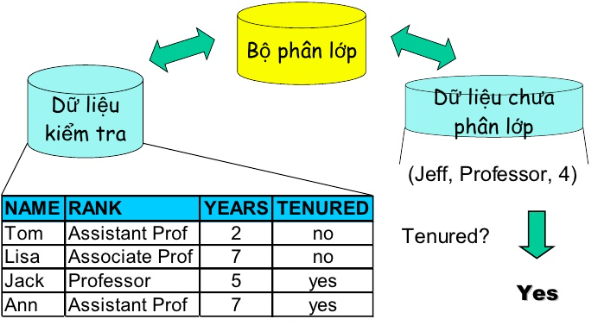
Bước 2: Vận hành mô hình

Phân lớp các đối tượng chưa biết :

Để tránh hiện tượng quá phù hợp (overfit) , một tập dữ liệu khác gọi là tập dữ liệu kiểm thử (testing set) sẽ được sử dụng để đo độ chính xác của giải thuật.Thông thường tập dữ liệu kiểm thử sẽ không chứa bất kỳ phần tử dữ liệu nào nằm trong tập huấn luyện.

Các nhãn của tập dữ liệu kiểm thử được dung để so sánh với nhãn được các giải thuật dự đoán. Tỷ lệ đoán đúng nhãn của các giải thuật phân lớp gọi là độ chính xác (accuracy) của giải thuật.

Khi chất lượng phân lớp của giải thuật là chấp nhận được trong một miền dữ liệu cụ thể nào đó, ta có thể dùng chúng để dự đoán lớp của các phần tử dữ liệu mới hoàn toàn chưa biết trước.



Hình 3-3: Mô hình phân lớp 2

* + 1. **Ứng dụng**

Chuẩn đoán bệnh trong y học,phát hiện tiền giả hay nghiên cứu thị trường.

* + 1. **Mô hình phân lớp**

Cây quyết định

Bộ phân lớp Bayesian

Mô hình phân lớp K-NN (K- nearest neighbor classifier)

Mạng Neural

Mạng Nơron

Giải thuật di truyền,..

* + 1. **Các tiêu chuẩn của mô hình phân lớp**

Trong ứng dụng cụ thể cần lựa chọn mô hình phân lớp phù hợp. Việc lựa chọn căn cứ vào sự so sánh các mô hình phân lớp với nhau, dựa trên tiêu chuẩn sau:

**Độ chính xác dự đoán**

Độ chính xác là khả năng của mô hình để dự đoán chính xác nhãn lớp của dữ liệu mới hay dữ liệu chưa biết.

**Tốc độ**

Tốc độ là những chi phí tính toán liên quan đến quá trình tạo ra và sử dụng mô hình

**Sức mạnh**

Sức mạnh là khả năng mô hình tạo ra những dự đoán đúng từ những dữ liệu nosie (gây nhiễu) hay dữ liệu với những giá trị thiếu.

**Khả năng mở rộng**

Khả năng mở rộng là khả năng thực thi hiệu quả trên lượng lớn dữ liệu của mô hình đã học.

**Tính hiểu được**

Tính hiểu được là mức độ hiểu và hiểu rõ những kết quả sinhh ra bởi mô hình đã học.

**Tính đơn giản**

Tính đơn giản là liên quan đến kích thước của cây quyết định hay độ động của các luật.

Trong các tiêu chuẩn trên khả năng mở rộng của mô hình phân lớp được nhấn mạnh và chú trọng phát triển , đặc biệt là cây quyết định.

* + 1. **Cây quyết định ứng dụng trong phân lớp**

**Cây quyết định :**

Gồm các nút trong biểu diễn giá trị thuộc tính, các nhánh biểu diễn đầu ra của kiểm tra, Nút lá biểu diễn nhãn lớp.

Cây được tạo theo hai giai đoạn là tạo cây và tỉa nhánh.

**Giai đoạn tạo cây:**

Bắt đầu tất cả các mẫu học đều nằm ở nút gốc. Sau đó các mẫu học được phân chia một cách đệ quy dựa trên thuộc tính được chọn.

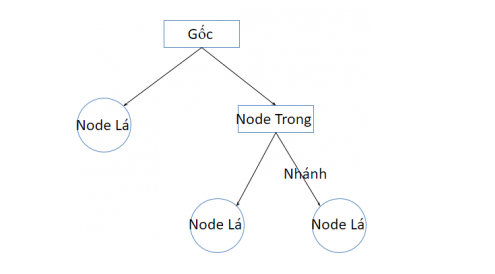
**Bước tỉa nhánh:**

Tìm và xóa những nhánh có phần tử không thể xếp vào lớp nào cả.

**Bước vận hành:**

Kiếm tra những giá trị thuộc tính của mẫu đối với các giá trị trên nhánh của cây.

Từ đó ta có dạng cây như sau:



Hình 3-4: Cấu trúc rẽ nhánh cây quyết định

Gốc: là node trên cùng của cây

Node trong: biểu diễn một kiểm tra trên một thuộc tính đơn (hình chữ nhật)

Nhánh: biểu diễn các kết quả của kiểm tra trên node trong (mũi tên)

Node lá: biểu diễn lớp hay sự phân phối lớp (hình tròn)

**Thuật toán tạo cây quyết định:**

**Bước 1**: Cây được xây dựng đệ quy từ trên xuống và theo cách chia để trị.

**Bước 2:** Ban đầu tất cả mẫu học đều nằm ở gốc.

**Bước 3**: Thuộc tính được phân loại (nếu là giá trị liên tục thì được rời rạc hóa)

**Bước 4:** Các mẫu học được phân chia đệ quy dựa trên thuộc tính chọn lựa.

**Bước 5:** Kiểm tra những thuộc tính được chọn dựa trên kinh nghiệm hoặc của một tiêu chuẩn thống kê.

**Điều kiện dừng phân chia tập học:**

Tất cả những mẫu học đối với một nút cho trước đều cùng lớp. Không còn thuộc tính nào để phân chia tiếp. Không còn mẫu học.

* + 1. **Ưu nhược điểm của cây quyết định**
* **Ưu điểm:** Cây quyết định là một thuật toán đơn giản và phổ biến. Thuật toán này được sử dụng rộng rãi bới những lợi ích của nó:
  + Mô hình sinh ra các quy tắc dễ hiểu cho người đọc, tạo ra bộ luật với mỗi nhánh lá là một luật của cây.
  + Dữ liệu đầu vào có thể là là dữ liệu missing, không cần chuẩn hóa hoặc tạo biến giả
  + Có thể làm việc với cả dữ liệu số và dữ liệu phân loại
  + Có thể xác thực mô hình bằng cách sử dụng các kiểm tra thống kê
  + Có khả năng là việc với dữ liệu lớn
* **Nhược điểm**: Kèm với đó, cây quyết định cũng có những nhược điểm cụ thể:
  + Mô hình cây quyết định phụ thuộc rất lớn vào dữ liệu của bạn. Thậm chí, với một sự thay đổi nhỏ trong bộ dữ liệu, cấu trúc mô hình cây quyết định có thể thay đổi hoàn toàn.
  + Cây quyết định hay gặp vấn đề overfitting
    1. **Thuật toán C4.5**

Tuy ID3 cũng sử dụng công thức Entropy và Information Gain để xây dựng mô hình cây quyết định có thể phân nhiều hơn 2 nhánh, tuy nhiên ID3 còn nhiều hạn chế không đem lại kết quả khi triển khai phân tích dữ liệu định lượng cụ thể là cây hồi quy (Regression tree). Chính vì thế ID3 không còn được sử dụng phổ biến.

C4.5 được xem là phiên bản nâng cấp của ID3

C4.5 là phiên bản của ID3 trên một số khía cạnh sau:

Trong bước xây dựng cây, chỉ tạo mô hình dựa trên các bản ghi đã xác định đầy đủ giá trị thuộc tính.

Trong bước vận hành cây quyết định, có thể phân loại những bản ghi có những giá trị thuộc tính chưa biết bằng việc ước lượng xác suất những kết quả có khả năng xảy ra.

**Giải mã của thuật toán C4.5**

Bắt đầu:

ComputerClassFrequency(T);

Kiểm tra các trường hợp cơ bản rồi tạo ra node quyết định

  if OneClass or FewCases

              return a leaf;

Create a decision node N;

Gán N.test = Thuộc tính với information gain được chuẩn hóa cao nhất

ForEach Attribute A

ComputeGain(A);

N.test=AttributeWithBestGain;

Thuật toán C4.5 sau đó sẽ lặp lại với các danh sách con nhỏ hơn và gán những node như là các node con

if (N.test is continuous)

find Threshold;

ForEach T’ in the splitting of T

If ( T’ is Empty )

Child of N is a leaf

else

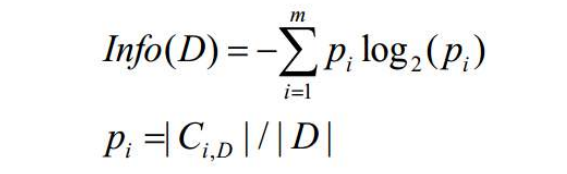
Child of N=FormTree(T’);

ComputeErrors of N;

return N

Công thức được sử dụng trong C4.5

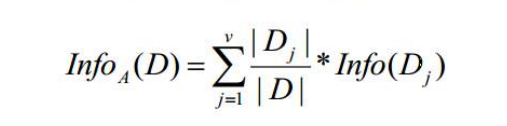
Lượng thông tin cần để phân loại một phần tử trong D (=Entropy của D) : Info(D)



Pi : xác suất để một phần tử bất kỳ trong D thuộc về lớp Ci  với i= 1…m

Ci,D : tập các phần tử Ci trong D

Lượng thông tin cần để phân loại một phần tử trong D dựa trên thuộc tính A: InfoA(D)



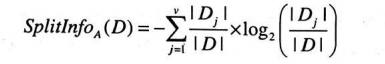
Thuộc tính A dùng phân tách D thahf v phân hoạch {D1 , D2,……., Dj,…. Dv}

Mỗi phân hoạch Dj gồm |Dj | phần tử trong D

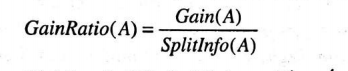
Lượng thông tin này sẽ cho biết mức độ trùng lặp giữa các phân hoạch, nghĩa là một phân hoạch chứa các phần tử từ một lớp hay nhiều lớp khác nhau.

Độ lợi thông tin (information gain). 

Information gain chính là độ sai biệt giữa trị thông tin Info(D) ban đầu và trị thông tin mới InfoA(D) (sau phân hoạch với A.)

Giá trị phân chia thông tin (split infomation). 

Tỉ số độ lợi





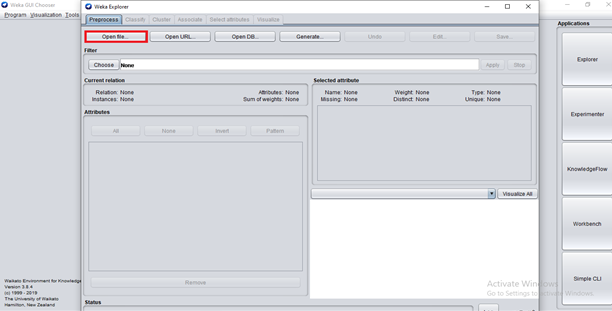
Split information A tương ứng với trị GainRatio(A) là trị lớn nhất.

Các thuộc tính có giá trị tỉ số độ lợi cao sẽ được chọn làm thuộc tính phân chia dữ liệu.

Nếu hàm SplitInfo(A) = 0 thì công thức trên không dùng được nên giá trị độ lợi thông tin của thuộc tính được chọn phải đủ lớn, ít nhất là lớn hơn giá trị trung bình độ lợi thông tin của tất cả thuộc tính.

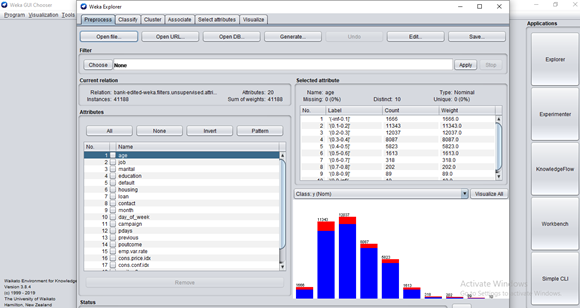
* + 1. **Sử dụng thuật toán C4.5 để phân lớp trong Weka**

Bước 1: Nhập file dữ liệu vào weka



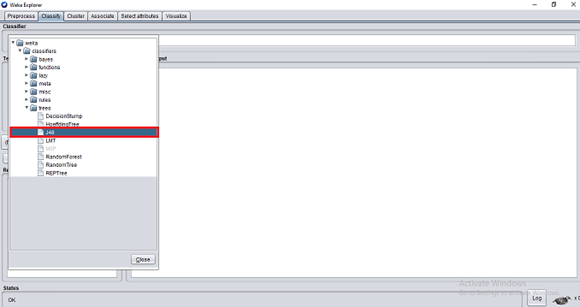
Hình 3-5: Đẩy file dữ liệu vào Weka

Bước 2: Giao diện weka sau khi nhập dữ liệu



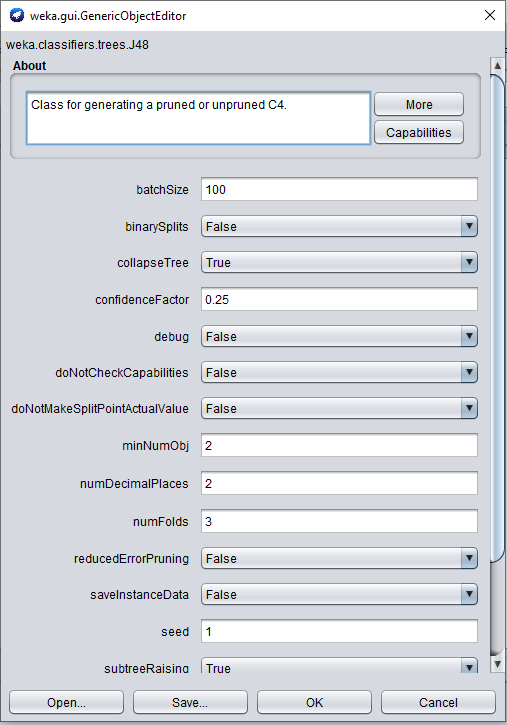
Hình 3-6: Màn hình hiển thị dữ liệu

Bước 3: Chọn thuật toán J48( C4.5 ) trong phân lớp cây quyết định



Hình 3-7: Chọn thuật toán j48 để phân lớp

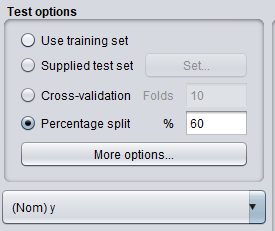
Bước 4: Chọn thông số

\

Hình 3-8: Điền thông số chạy thuật toán

|  |  |
| --- | --- |
| Tham số | Mô tả |
| batchSize | Số lượng mẫu dữ liệu trong một batch. Vì dữ liệu đã được chia thành các phần (number of batch), mỗi phần số kích thước là batch size. |
| binarySplits | Có sử dụng phân tích nhị phân ở trong các thuộc tính nominal khi xây dựng cây hay không? |
| collapseTree | Có các thành phần được bỏ đi mà không giảm bớt lỗi |
| confidenceFactor | Mức tin tưởng để quyết định xem có xét tỉa hay không |
| debug | Nếu thiết lập là true, sự phân lớp có thể xảy ra các thông tin bổ sung tới màn console |
| minNumObj | Số lượng nhỏ nhất các thể hiện trên mỗi nút lá |
| numFolds | Xác định rõ lượng dữ liệu sử dụng cho việc giảm bớt lỗi xét tỉa. Mỗi fold được sử dụng cho việc xét tỉa, phần còn lại sử dụng cho việc phát triển cây |
| reducedErrorPruning | Có xem tỉa giảm bớt lỗi được sử dụng thay thế xén tỉa C4.5 |

Những lựa chọn để test



Hình 3-9: Các lựa chọn tập kiểm thử

|  |  |
| --- | --- |
| Tham số | Mô tả |
| Use training set | Dùng toàn bộ tập dữ liệu để xây dựng mô hình và dùng toàn bộ tập dữ liệu để test |
| Supplied test set | Dùng toàn bộ tập dữ liệu đang mở để xây dựng mô hình và dùng tập dữ liệu khác để test |
| Cross-validation | Đánh giá chéo, Folds thể hiện tập dữ liệu sẽ bị chia ra làm X với số lượng phần tử bằng nhau, sẽ đưa từng tập vào huấn luyện những tập còn lại sẽ dung để kiểm tra |
| Percentage split | Phân tách dữ liệu thành 2 phần X% dữ liệu sẽ dùng để huấn luyện và còn lại để kiểm tra |

Bước 4: Kết quả

Thực nghiệm sẽ tiến hành 05 lần với các tỷ lệ như sau:

L1 : Sử dụng J48 cắt tỉa trên tập dữ liệu với tỷ lệ huấn luyện 50%

L2 : Sử dụng J48 cắt tỉa trên tập dữ liệu với tỷ lệ huấn luyện 55%

L3 : Sử dụng J48 cắt tỉa trên tập dữ liệu với tỷ lệ huấn luyện 60%

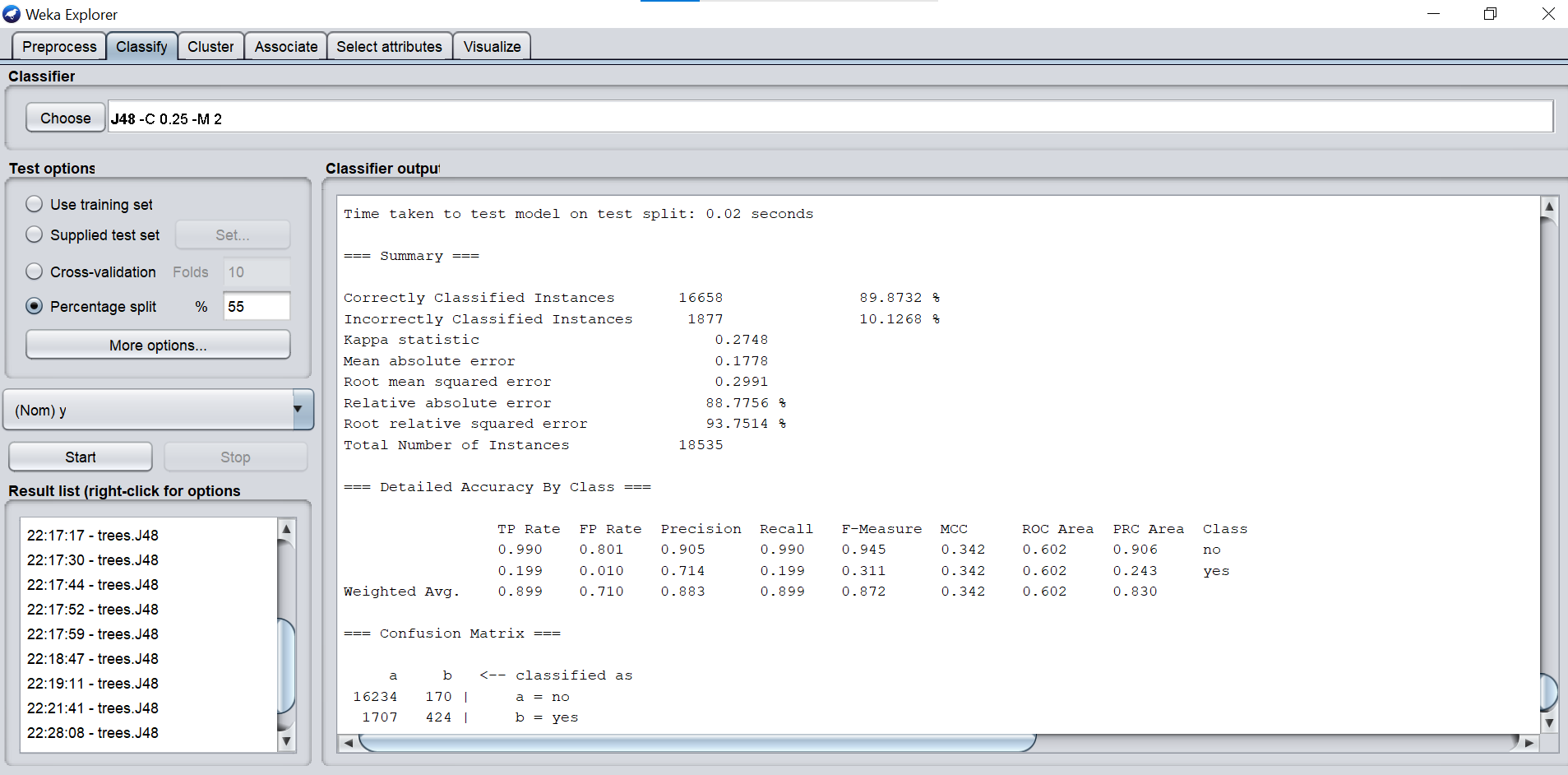
L4 : Sử dụng J48 cắt tỉa trên tập dữ liệu tỷ lệ huấn luyện 65%

L5 : Sử dụng J48 cắt tỉa trên tập dữ liệu tỷ lệ huấn luyện 70%

Kết quả các lần chạy ta có bảng sau:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Lần chạy | Tỷ lệ huấn luyện /kiểm thử | Tổng số bản ghi | Tổng số bản ghi đưa vào phân lớp | Tổng số thuộc tính | Số mẫu phân lớp đúng | Số mẫu phân lớp sai | Tỷ lệ phân lớp đúng | Tỷ lệ phân lớp sai |
| 1 | 50/50 | 41188 | 20594 | 20 | 18507 | 2087 | 89.866 % | 10.134 % |
| 2 | 55/45 | 41188 | 18535 | 20 | 16658 | 1877 | 89.8732 % | 10.1268 % |
| 3 | 60/40 | 41188 | 16475 | 20 | 14792 | 1683 | 89.7845 % | 10.2155 % |
| 4 | 65/35 | 41188 | 14416 | 20 | 12935 | 1481 | 89.7267 % | 10.2733 % |
| 5 | 70/30 | 41188 | 12356 | 20 | 11065 | 1291 | 89.5516 % | 10.4484 % |

Từ các lần chạy thực nghiệm đã thực hiện, lần chạy lần 1 đạt hiệu quả phân lớp 89.8732 % cao nhất trong 5 lần thử. Như vậy lần chạy 1 với tỷ lệ huấn luyện 55% là lựa chọn phù hợp nhất.



Hình 3-10: Kết quả thuật toán J48

Kết quả khi lựa chọn Percentage split để test với 55% dữ liệu dùng để test 45% còn lại dung để kiểm tra.

Từ hình trên ta thấy số các bản ghi (đối tượng) được phân lớp đúng là 16658 chiếm gần 90% và số các bản ghi phân lớp sai là 1877 chiếm gần 10%.Tổng số đối tượng đưa vào phân lớp là 18535 bản ghi.

Ngoài ra có các thống kê như :

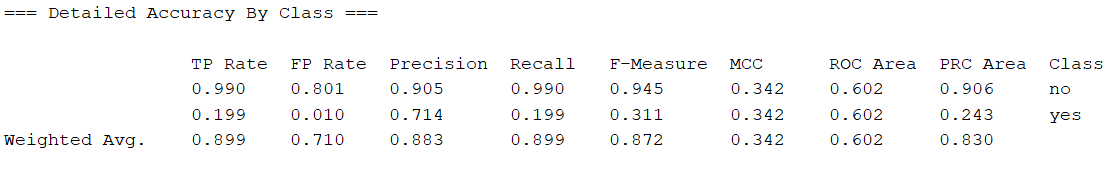
- Kappa statistic là một thống kê giúp đo đạc độ đồng thuận giữa các thành phần định tính (phân loại).

- Mean absolute error là độ lỗi trung bình tuyệt đối, sự khác biệt giữa các ước lượng và những gì được đánh giá.

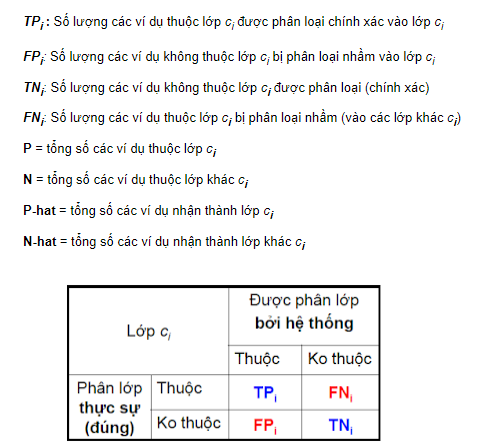
- Root mean squared error là độ lệch trung bình bình phương gốc hoặc sai số trung bình bình phương gốc là thước đo được sử dụng thường xuyên về sự khác biệt giữa các giá trị được dự đoán bởi mô hình hoặc công cụ ước tính và các giá trị được quan sát.

- Relative absolute error là sự khác biệt giữa một giá trị chính xác và một số xấp xỉ với nó.

Detailed Accuracy By Class (Độ chính xác chi tiết theo lớp)



**Các độ đo hay dùng**



**Ý nghĩa các giá trị quan trọng:**

TPrate (tỷ lệ mẫu tích cực - Positive):



FPrate (tỷ lệ mẫu tích cực sai):



Precision (Độ chính xác):



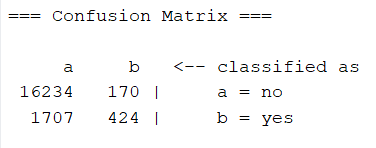
Recall (Độ hồi tưởng):



F- measure (Độ đo tổng hợp)



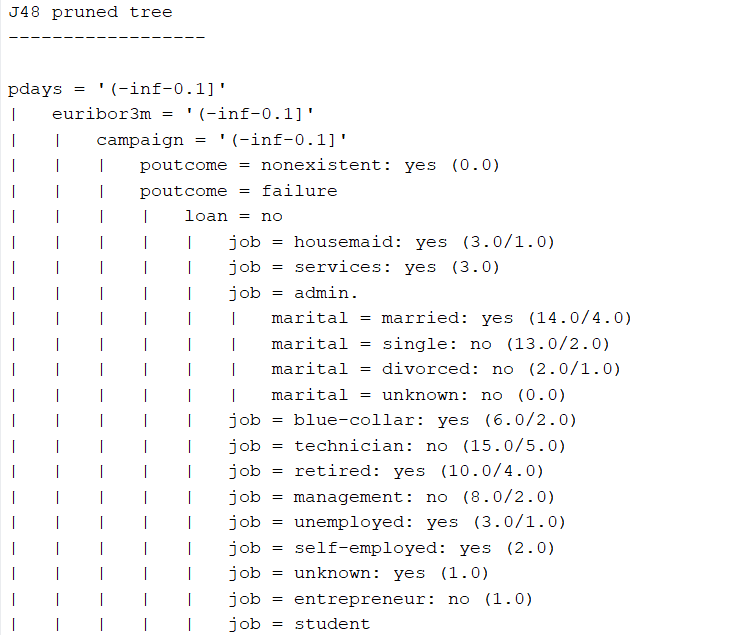
Bảng thông số kết quả chạy thuật toán:



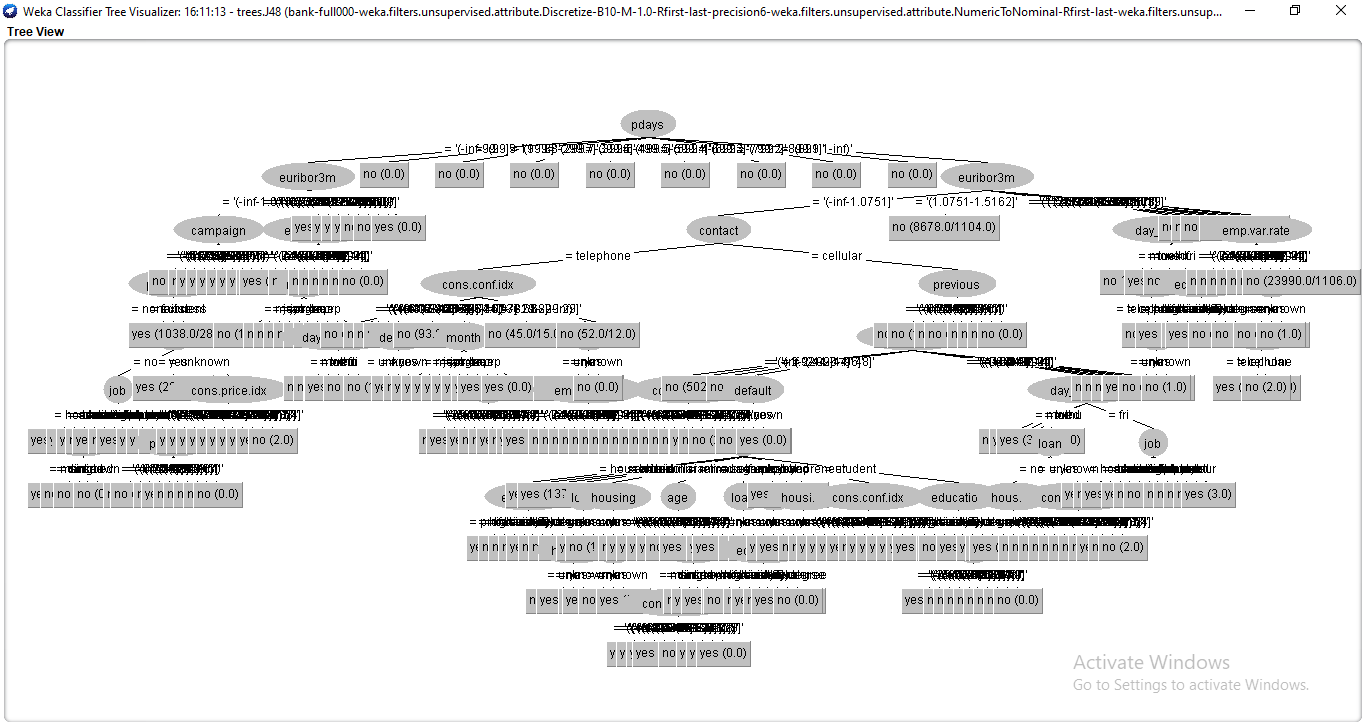
Có 16234 mẫu vào đúng lớp a = no và 170 mẫu vào đúng lớp b = yes.

Có 1707 mẫu vào sai lớp a = no và 424 mẫu vào sai lớp b = yes.

**Dạng cây quyết định tạo ra từ tập dữ liệu thử nghiệm**



Hình 3-11: Kết quả dạng cây của thuật toán J48





Hình 3-12: Cây của thuật toán J48

Cây sinh ra có 304 lá và kích thước là 356

Kết quả cây như sau:

pdays = '(99.9-199.8]': no (0.0)

pdays = '(199.8-299.7]': no (0.0)

pdays = '(299.7-399.6]': no (0.0)

pdays = '(399.6-499.5]': no (0.0)

pdays = '(499.5-599.4]': no (0.0)

pdays = '(599.4-699.3]': no (0.0)

pdays = '(699.3-799.2]': no (0.0)

pdays = '(799.2-899.1]': no (0.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = nonexistent: yes (0.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = housemaid: yes (3.0/1.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = services: yes (3.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin -> marital = married: yes (14.0/4.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin -> marital = single: no (13.0/2.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin -> marital = divorced: no (2.0/1.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin-> marital = unknown: no (0.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = blue-collar: yes (6.0/2.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = technician: no (15.0/5.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = retired: yes (10.0/4.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = management: no (8.0/2.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = unemployed: yes (3.0/1.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = self-employed: yes (2.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = unknown: yes (1.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = entrepreneur: no (1.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(-inf-0.7]': no (0.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(0.7-1.4]': no (0.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(1.4-2.1]': no (10.0/5.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(2.1-2.8]': no (0.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(2.8-3.5]': yes (2.0)

pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(3.5-4.2]': no (4.0)

v.v…..

* 1. **Luật kết hợp** 
     1. **Cơ sở lý thuyết**

Được giới thiệu từ năm 1993, bài toán khai thác luật kết hợp nhận được rất nhiều sự quan tâm của nhiều nhà khoa học. Ngày nay việc khai thác các luật như thế vẫn là một trong những phương pháp khai thác mẫu phổ biến nhất trong việc khám phá tri thức và khai thác dữ liệu.

Mục đích chính của khai phá dữ liệu là các tri thức được kết xuất ra sẽ được sử dụng trong dự báo thông tin trợ giúp trong sản xuất kinh doanh và nghiên cứu khoa học.

Trong hoạt động sản xuất kinh doanh, ví dụ kinh doanh các mặt hàng tại siêu thị, các nhà quản lý rất thích có được các thông tin mang tính thống kê. Những thông tin như vậy rất hữu ích trong việc định hướng kinh doanh. Vậy vấn đề đặt ra là liệu có tìm được các luật bằng các công cụ khai phá dữ liệu hay không? Câu trả lời hoàn toàn có thể. Bởi đó chính là nhiệm vụ của luật kết hợp.

* + 1. **Định nghĩa**

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.

* ***Định nghĩa luật kết hợp****:*

Một luật kết hợp là một quan hệ có dạng X ⇒ Y, với X, Y ⊂I là các tập mục gọi là itemsets, và X ∩Y = ∅ . X là tiền đề, Y là mệnh đề kết quả.

***=>> Ý nghĩa:***

Khi xuất hiện X thì Y cũng xuất hiện (với xác xuất nào đó)

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.

* ***Định nghĩa độ hỗ trợ***

Định nghĩa 1: 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)

S0 = |{T ⊂ D:Y ⊃ X}|

| D|

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

Định nghĩa 2: 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).

Supp(X→Y) = | { T ⊂ D T ⊇ X ∪Y} |

|D|

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.

* ***Định nghĩa độ tin cậy***

*Định nghĩa 1:* 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:

Supp(X→Y)=P(X∪ Y)

Conf (X→Y) = P(Y/X)=supp(X∪Y)/supp(X)

*Định nghĩa 2:* 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.

=>> Luật kết hợp thường được đánh giá dựa trên 2 độ đo là độ hỗ trợ và độ tin cậy

* ***Định nghĩa 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: 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.

* + 1. **Một số thuật toán phát hiện luật kết hợp**
       1. ***Thuật toán Apriori***

Thuật toán dựa trên một nhận xét khá đơn giản là bất kỳ tập hợp con nào của tập xuất hiện σ thường xuyên cũng là tập xuất hiện σ–thường xuyên. Do đó, trong quá trình đi tìm các tập ứng cử viên, nó chỉ cần dùng đến các tập ứng cử viên vừa xuất hiện ở bước ngay trước đó, chứ không cần dùng đến tất cả các tập ứng cử viên (cho đến thời điểm đó). Nhờ vậy, bộ nhớ được giải phóng đáng kể.

Bước 1: Cho trước ngưỡng độ hỗ trợ 0 ≤ σ ≤ 1. Tìm tất cả các mặt hàng xuất hiện σ– thường xuyên.

Bước 2: Ta tiến hành ghép đôi các phần tử của L1 (không cần để ý đến thứ tự), được tập C2, tạp gọi là tập các ứng cử viên có 2 phần tử. Sở dĩ chỉ gọi là “ứng cử viên”, vì chưa chắc chúng đã là σ– thường xuyên. Sau khi kiểm tra (dùng định nghĩa), ta lọc ra được các tập hợp σ– thường xuyên có 2 phần tử. Ký hiệu tập hợp này là L2.

Bước 3: Với chủ ý đã nêu (về tính chất tăng dần của các tập hợp σ– thường xuyên), ta tiến hành tìm các ứng cử viên có 3 phần tử (lấy từ L1). Gọi nó là tập C3. Lưu ý là nếu {A, B, C} muốn là “ứng cử viên” thì các tập 2 phần tử {A, B},{B,C},{C, A } đều phải là σ – thường xuyên, tức là chúng đều là phần tử của tập L2. Ta đi “kiểm tra tư cách đại biểu” trong tập C3 và lọc ra được tập các tập hợp σ– thường xuyên có 3 phần tử. Tập hợp này được ký hiệu là L3.

Bước 4: Ta tiến hành tìm các ứng cử viên có n phần tử. Gọi tập của chúng là tập Cn và từ đây, lọc ra Ln là tập tập các tập hợp σ– thường xuyên có n phẩn tử. Cốt lõi của thuật toán Apriori là hàm apriori\_gen() do Agrawal đề nghị năm 1994. Hàm này hoạt động theo 2 bước, bước 1- tập hợp Lk-1 tự kết nối (join) với chính nó để tạo ra tập ứng cử viên Ck. Sau đó hàm apriori\_gen() loại bỏ các tập hợp có một hợp con (k-1) phần tử không nằm trong Lk-1 (vì chúng không thể là tập hợp xuất hiện σ– thường xuyên, theo như nhận xét ban đầu).

***=>>Ý tưởng***

Tạo các tập 1\_itemset: từ các item trên CSDL, ta xác định độ hỗ trợ s cho từng item dựa vào CSDL đã mã hóa, loại đi các item có s < minsup.

Tạo các tập 2\_itemset: xác định độ hỗ trợ s cho tập gồm 2 item, loại đi các item có s < minsup. …

Tạo các tập k\_itemset: xác định độ hỗ trợ s cho tập gồm k item, loại đi các item có s < misup.

Thuật giải: có 2 bước chính

B1: Sinh ra tập Itemset phổ biến.

B2: Tìm ra luật.

Apriori dùng để giảm các thuộc tính, loại bỏ các thuộc tính không cần thiết.

Apriori cần 2 tham số là minsup và minconf, minsup dùng để sinh ra tập các itemsets phổ biến còn minconf dùng để tìm luật.

Input: CSDL giao dịch D, ngưỡng minsup.

Output: Các tập phổ biến

Procedure Apriori

1. Quét CSDL D để tìm các tập mục phổ biến 1\_itemset, L1.

2. for(k=2; L k-1≠ ∅; k++){

3. Ck=Apriori\_Gen(L k-1, minsup); //sinh ứng cử k-itemset

4. for(mỗi một tác vụ t∈D) { //quét D để đếm số lần xuất hiện của các tập mục trong Ck trong cơ sở dữ liệu

5. Ct=subnet(Ck, t);//lấy tập các ứng cử trong Ck mà nó là tập con của t

6. for(mỗi một ứng cử c∈Ct)

7. c.count++; }

8. Lk={c∈Ck | c.count ≥ minsup}

9. return L = Uk i=1 Li

=>> Đánh giá giải thuật Apriori các yếu tố ảnh hưởng:

- Lựa chọn giá trị ngưỡng minsup: Giá trị minsup quá thấp sẽ sinh ra nhiều tập mục thường xuyên. Điều này sẽ làm tăng số lượng tập mục phải xét.

- Số lượng các mục trong cơ sở dữ liệu (các giao dịch): Cần thêm bộ nhớ để lưu giá trị độ hỗ trợ với mỗi mục. Nếu số lượng các mục(tập mục mức 1) thường xuyên tăng lên thì chi phí và chi phí I/O (duyệt các giao dịch) cũng tăng.

- Kích thước của cơ sở dữ liệu (các giao dịch): Giải thuật phải duyệt cơ sở dữ liệu nhiều lần, do đó chi phí tính toán của Apriori tăng lên khi số lượng các giao dịch tăng lên.

- Kích thước trung bình của các giao dịch: Khi kích thước (số lượng các mục) trung bình của các giao dịch tăng lên, thì độ dài tối đa của các tập mục thường xuyên cũng tăng.

* + - 1. **Thuật giả FP\_Growth**

FP\_Growth sử dụng một cấu trúc dữ liệu gọi là FP\_tree (Frequent Pattern tree)

FP\_tree là một thể hiện cô đọng các thông tin có liên quan đến thông tin thể hiện tính thường xuyên của các tập mục trong CSDL

Mỗi nhánh của cây FP\_tree thể hiện một tập mục phổ biến, và các nút dọc theo các nhánh được lưu trữ theo thứ tự giảm dần của tính phổ biến tương ứng với các mục, các mục ở lá của cây có tính phổ biến thấp nhất.

Cây FP\_tree có một bảng header kết hợp với nó. Bảng header lưu các mục cùng với số lần xuất hiện của nó trong CSDL theo thứ tự giảm dần của tính phổ biến (mỗi mục chiếm một dòng của bảng.

Mỗi mục của bảng chứa một nút đầu danh sách liên kết với tất cả các nút của cây FP\_tree mà nút đó có tên trùng với tên của nó.

FP\_gowth chỉ duyệt CSDL 2 lần để khai phá tất cả các tập mục phổ biến. Lần 1 để xác định tần xuất của từng tập mục trong CSDL. Lần 2 để xây dựng cây FP\_t

**=>> Ý tưởng:**

FP-Growth biểu diễn dữ liệu của các giao dịch bằng một cấu trúc dữ liệu gọi là FP –tree.

FP-Growth sử dụng cấu trúc FP-tree để xác định trực tiếp các tập mục thường xuyên.

Biểu diễn bằng FP-tree:

- Với mỗi giao dịch, FP-tree xây dựng một đường đi (path) trong cây.

- Hai giao dịch có chứa cùng một số mục, thì đường đi của chúng sẽ có phần (đoạn) chung. Càng nhiều các đường đi có các phần chung, thì việc biểu diễn bằng FP-tree sẽ càng gọn.

- Nếu kích thước của FP-tree đủ nhỏ có thể lưu trữ trong bộ nhớ làm việc, thì giải thuật FP-Growth có thể xác định các tập thường xuyên trực tiếp từ FP-tree lưu trong bộ nhớ.

Xây dựng FP-tree:

- Ban đầu, FP-tree chỉ chứa duy nhất nút gốc (được biểu diễn bởi ký hiệu null).

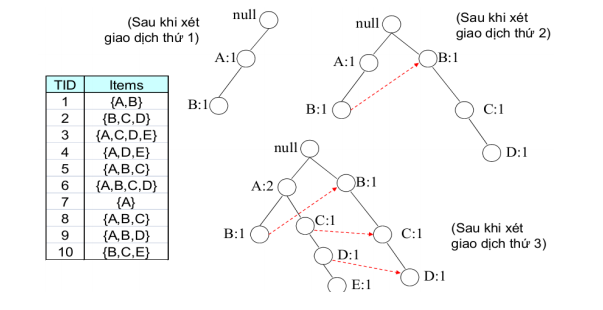
- CSDL các giao dịch được duyệt lần thứ 1, để xác định (tính) độ hỗ trợ của mỗi mục.

- Các mục không thường xuyên bị loại bỏ.

- Các mục thường xuyên được sắp xếp theo thứ tự giảm dần về độ hỗ trợ.

- CSDL các giao dịch được duyệt lần thứ 2, để xậy dựng FP-tree.

Ví dụ:

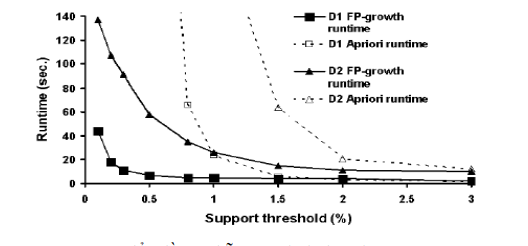


Sinh các tập mục thường xuyên:

- FP-Growth sinh các tập mục thường xuyên trực tiếp từ FP-tree từ mức lá đến mức gốc (bottom-up).

- Vì mỗi giao dịch được biểu diễn bằng một đường đi trong FP -tree, chúng ta có thể xác định các tập mục trong FPtree, chúng ta có thể xác định các tập mục thường xuyên kết thúc bởi một mục (vd: E), bằng cách duyệt các đường đi chứa mục đó (E)

**3.2.3.3. So sánh giữa giải thuật Apriori và giải thuật FP-Growth**



Hình 3-13: Biểu đồ: Độ hỗ trợ - Thời gian chạy

=>> Các cải tiến của giải thuật Apriori

Kỹ thuật dựa trên bảng băm (hash-based technique): Một k-itemset ứng với hashing bucket count nhỏ hơn minimum support threshold không là một frequent itemset.

Giảm giao dịch (transaction reduction): Một giao dịch không chứa frequent k-itemset nào thì không cần kiểm tra ở các lần sau (cho k+1-itemset).

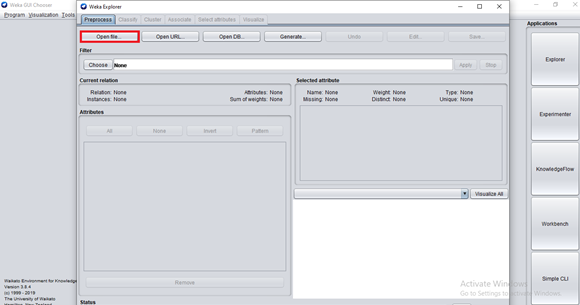
Phân hoạch (partitioning): Một itemset phải frequent trong ít nhất một phân hoạch thì mới có thể frequent trong toàn bộ tập dữ liệu.

Lấy mẫu (sampling): Khai phá chỉ tập con dữ liệu cho trước với một trị support threshold nhỏ hơn và cần một phương pháp để xác định tính toàn diện (completeness).

Đếm itemset động (dynamic itemset couting): Chỉ thêm các itemset dự tuyển khi tất cả các tập con của chúng được dự đoán là frequent.

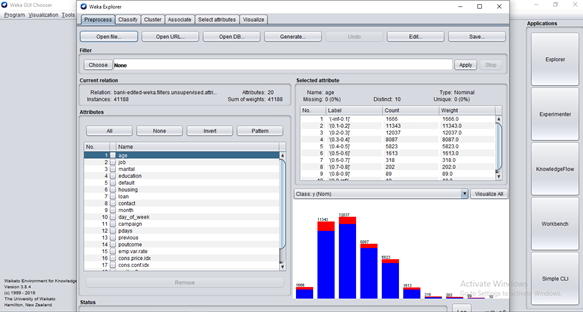
* + 1. **Các bước thực hiện khai phá luật kết hợp và ý nghĩa thông số**
       1. **Thuật toán****Apriori**

Bước 1: Nhập file dữ liệu vào weka



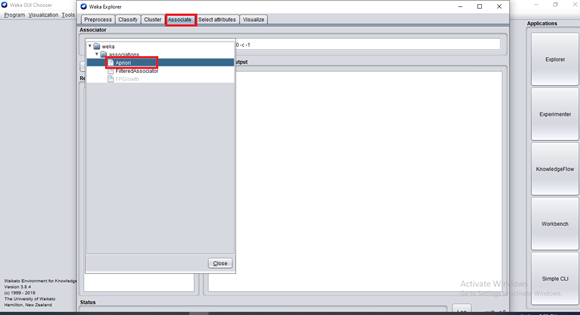
Hình 3-14: Đẩy file dữ liệu vào Weka

Bước 2: Giao diện weka sau khi nhập dữ liệu



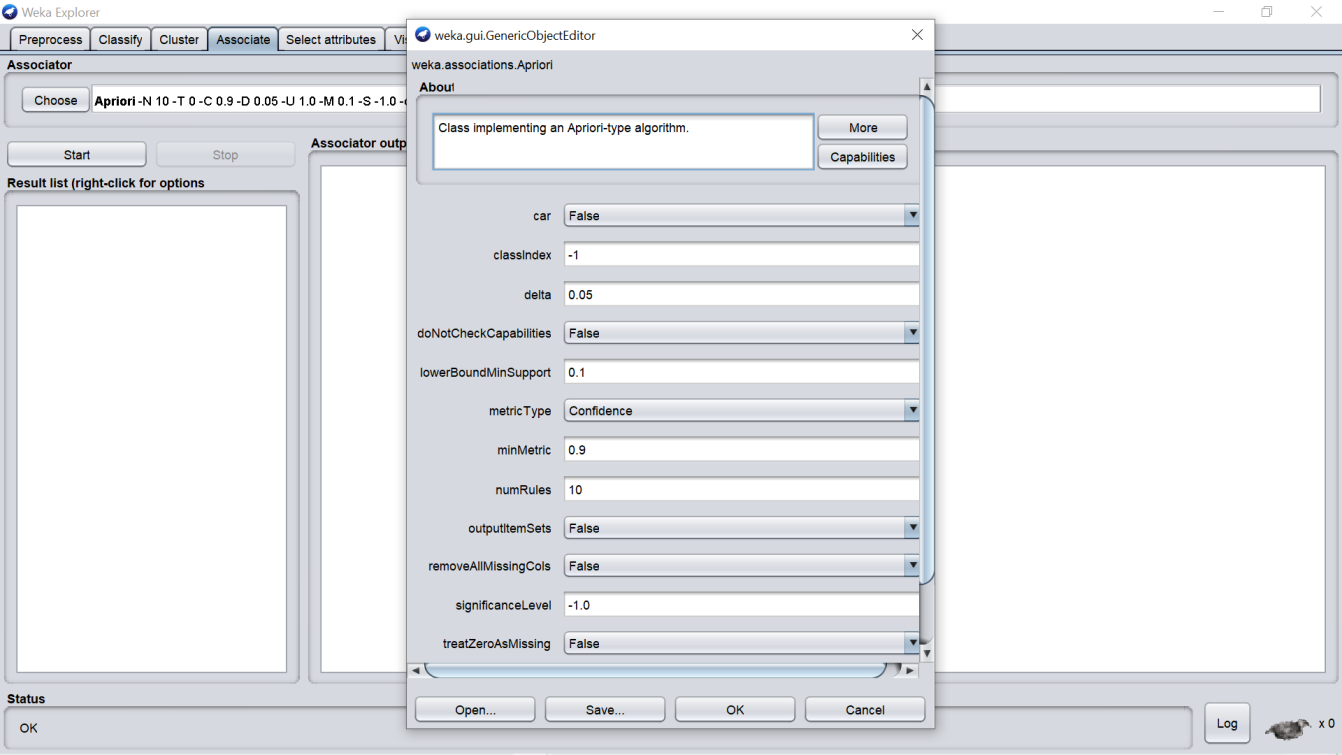
Hình 3-15: Màn hình hiển thị dữ liệu đầu vào

Bước 3: Chọn luật kết hợp Apiori



Hình 3-16: Chọn luật kết hợp là Apiori

Bước 4: Chọn thông số



Hình 3-17: Chọn thông số cho thuật toán Apiori

***Giải thích thông số:***

*Car:* Khai phá luật kết hợp phân lớp

*Classindex:* -1 ở đấy có nghĩa là lấy lớp cuối cùng

*lowerBoundMinSupport:* Cận dưới của minimum support ( cận dưới của độ hỗ trợ tối thiếu).

*metricType:* Có 4 loại metricType là Confidence, Lift , Leverage và Conviction là dạng thang đo độ tin cậy của luật kết hợp.

*OutputltemSets:* Hiển thị tập dữ liệu.

*RemoveAllMissingCols:* loại bỏ các cột không chứa giá trị.

*SignificanceLevel*: mức ý nghĩa, chỉ hoạt động với metric type là Confidence

*TreatZeroAsMissing:* loại bỏ giá trị đầu tiên mỗi row

*UpperBoundMinSupport:* cận trên độ hỗ trợ tối thiểu

*Verbose:* chạy chế độ hiển thị chi tiết quá trình

*Minimum metric score*: Chỉ quan quan đến các luật có metric score cao hơn giá trị này.

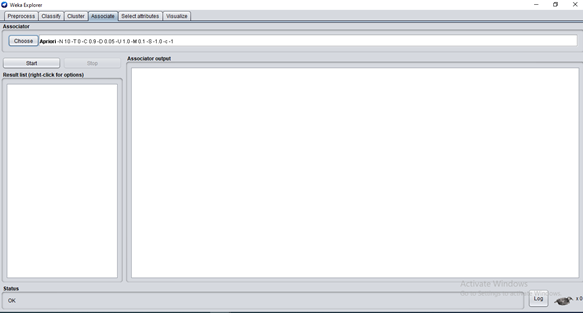
*numRules:* Số luật muốn tìm (các luật sẽ được sắp xếp theo thứ tự giảm dần của metric score.

*significanceLevel:* Mức ý nghĩa (chỉ dùng khi metricType là confidence).

upperBoundMinSupport: Cận trên của minimum support (bắt đầu lặp lại việc giảm minimum support từ upperBoundMinSupport đến lowerBoundMinSupport).

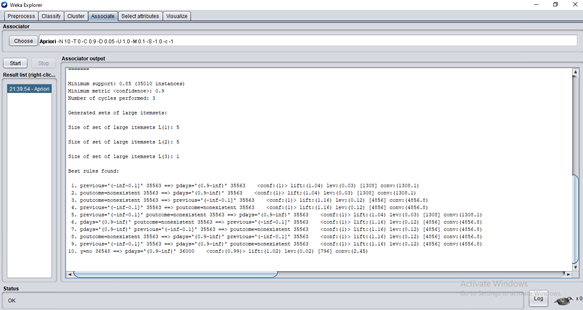
Bước 5: Chạy chương trình

Giả sử ta chọn thông số như sau:



Hình 3-18: Nhìn lại thông số đã chọn cho thuật toán Apiori

Khi đó ta được kết quả của 10 luật tốt nhất là:



Hình 3-19: Kết quả của thuật toán Apiori

Bước 6 : Đọc kết quả

1. previous='(-inf-0.1]' 35563 ==> pdays='(0.9-inf)' 35563 <conf:(1)> lift:(1.04) lev:(0.03) [1308] conv:(1308.1)

=>> *Với thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1 ( 35563 trường hợp) =>> pday > 0.9 ( 35563 trường hợp) (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước với độ tin cậy là 100%.*

2. poutcome=nonexistent 35563 ==> pdays='(0.9-inf)' 35563 <conf:(1)> lift:(1.04) lev:(0.03) [1308] conv:(1308.1)

*=>> Với thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) =>> pday > 0.9 ( 35563 trường hợp) (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước với độ tin cậy là 100%.*

3. poutcome=nonexistent 35563 ==> previous='(-inf-0.1]' 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

*=>> Với thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) =>> previous ( số cuộc gọi được thực hiện) <= 0.1 ( 35563 trường hợp) với độ tin cậy là 100%.*

4. previous='(-inf-0.1]' 35563 ==> poutcome=nonexistent 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

=>> *Với thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1 ( 35563 trường hợp) =>> thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) với độ tin cậy là 100%.*

5. previous='(-inf-0.1]' poutcome=nonexistent 35563 ==> pdays='(0.9-inf)' 35563 <conf:(1)> lift:(1.04) lev:(0.03) [1308] conv:(1308.1)

=>> *Với thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1 và thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) =>>pday > 0.9 ( 35563 trường hợp) (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước với độ tin cậy là 100%.*

6. pdays='(0.9-inf)' poutcome=nonexistent 35563 ==> previous='(-inf-0.1]' 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

=>> *pday > 0.9 số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước =>> thuộc tính previous ( số cuộc gọi được thực hiện) ( 35563 trường hợp) với độ tin cậy là 100%.*

7. pdays='(0.9-inf)' previous='(-inf-0.1]' 35563 ==> poutcome=nonexistent 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

=>> *Với pday > 0.9 số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1 =>> thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) với độ tin cậy là 100%.*

8. poutcome=nonexistent 35563 ==> pdays='(0.9-inf)' previous='(-inf-0.1]' 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

=>> *Với thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) =>> pday > 0.9 số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1 ( 35563 trường hợp) với độ tin cậy là 100%.*

9. previous='(-inf-0.1]' 35563 ==> pdays='(0.9-inf)' poutcome=nonexistent 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

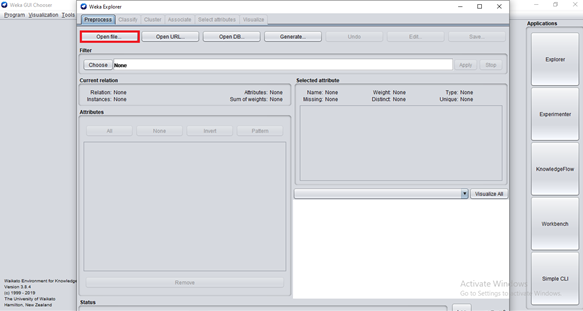
=>> *Với thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1 ( 35563 trường hợp) =>> pday > 0.9 số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) với độ tin cậy là 100%.*

10. y=no 36548 ==> pdays='(0.9-inf)' 36000 <conf:(0.99)> lift:(1.02) lev:(0.02) [796] conv:(2.45)

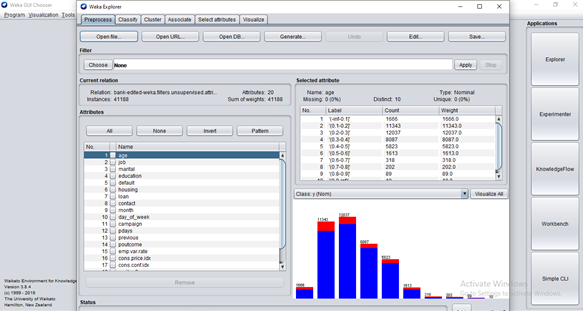
=>> *Nếu thuộc tính phân lớp = 0 ( 36548 trường hợp ) =>> pday > 0.9 số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và thuộc tính previous ( số cuộc gọi được thực hiện) với độ tin cậy là 99%*

* + - 1. **Thuật toán FB\_ Growth**

Bước 1: Nhập file dữ liệu vào weka

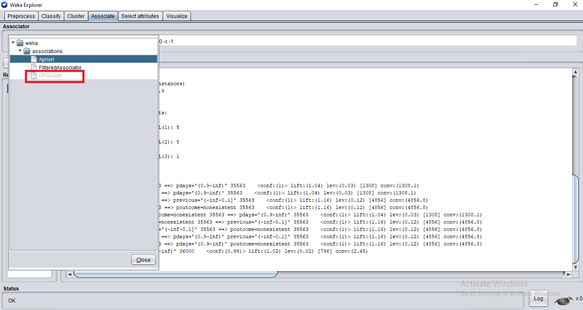


Hình 3-20: Đẩy file dữ liệu vào Weka

Bước 2: Giao diện weka sau khi nhập dữ liệu

Hình 3-21: Giao diện weka sau khi nhập dữ liệu

Bước 3: Chọn luật kết hợp FB\_ *Growth*



Hình 3-22: Chọn thuật toán FB\_Growth

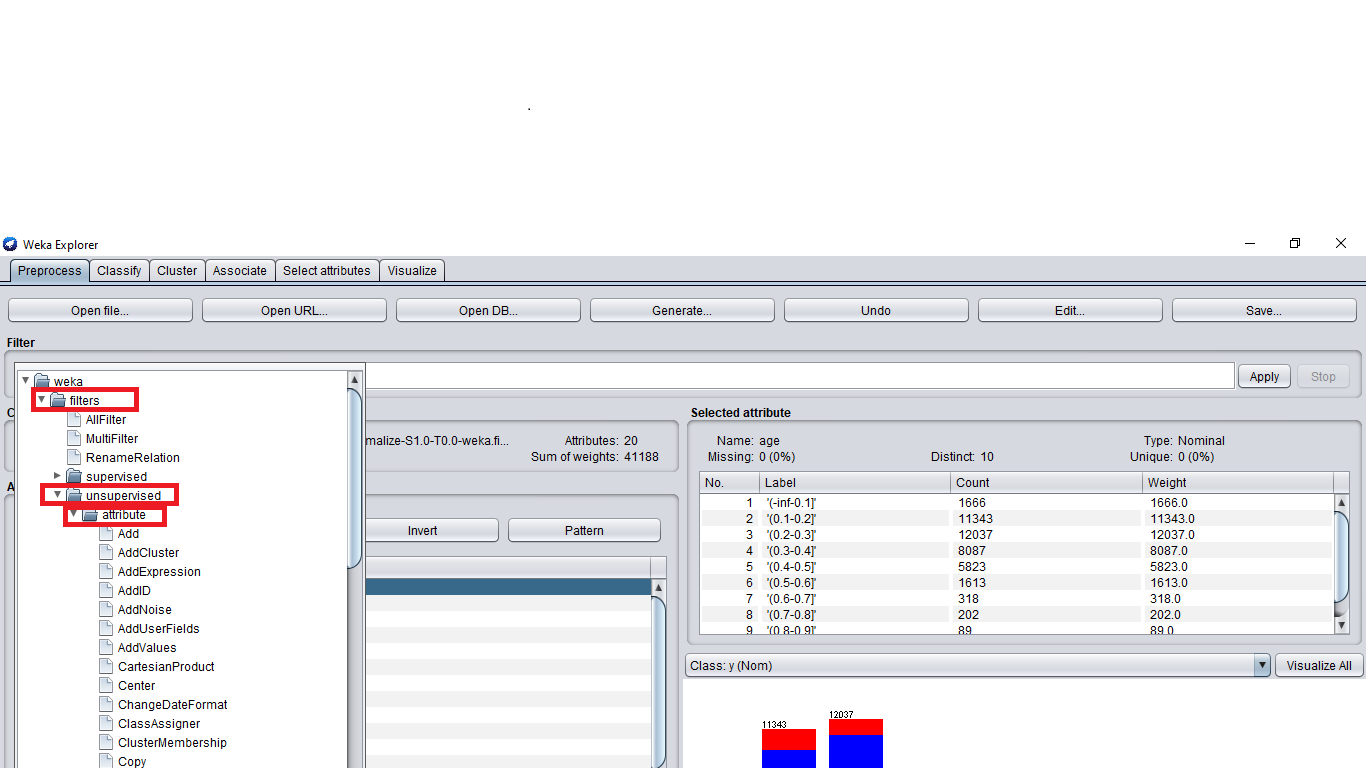
Ta không chọn được luật kết hợp FB\_ *Growth*

FB\_ *Growth* chỉ làm việc với dữ liệu nhị phân

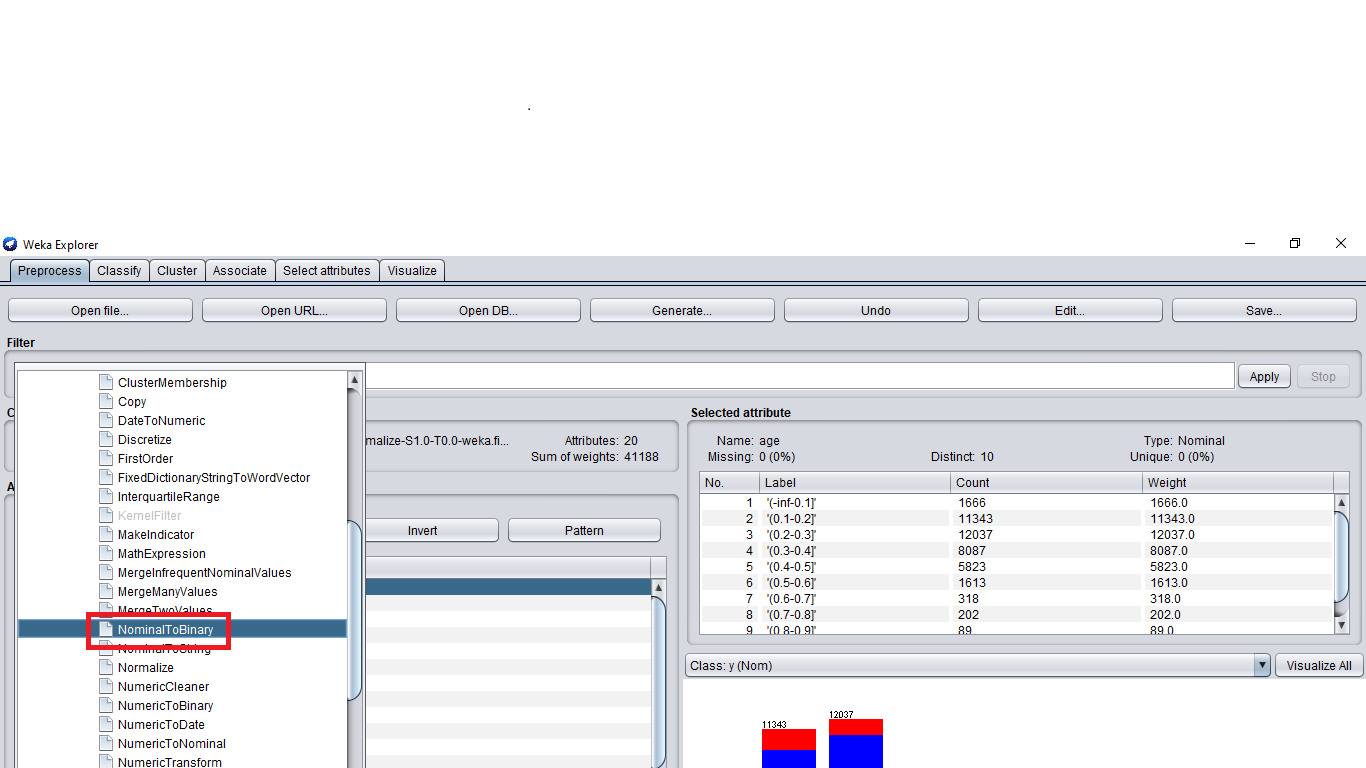
Do cần thực hiện chuyển đổi dữ liệu trước khai phá

Phương thức chuyển đổi là: NominaltoBinary filter

Bước 4: Chuyển đổi dữ liệu

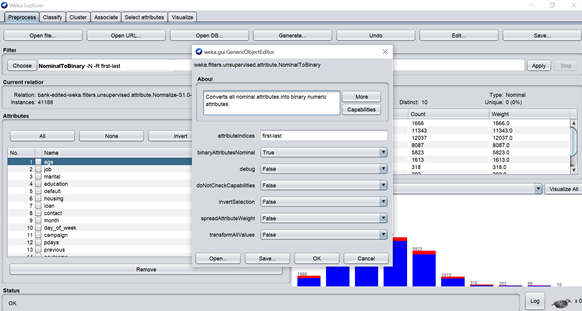


Hình 3-23: Chuyển đổi định dạng thành nhị phân

**

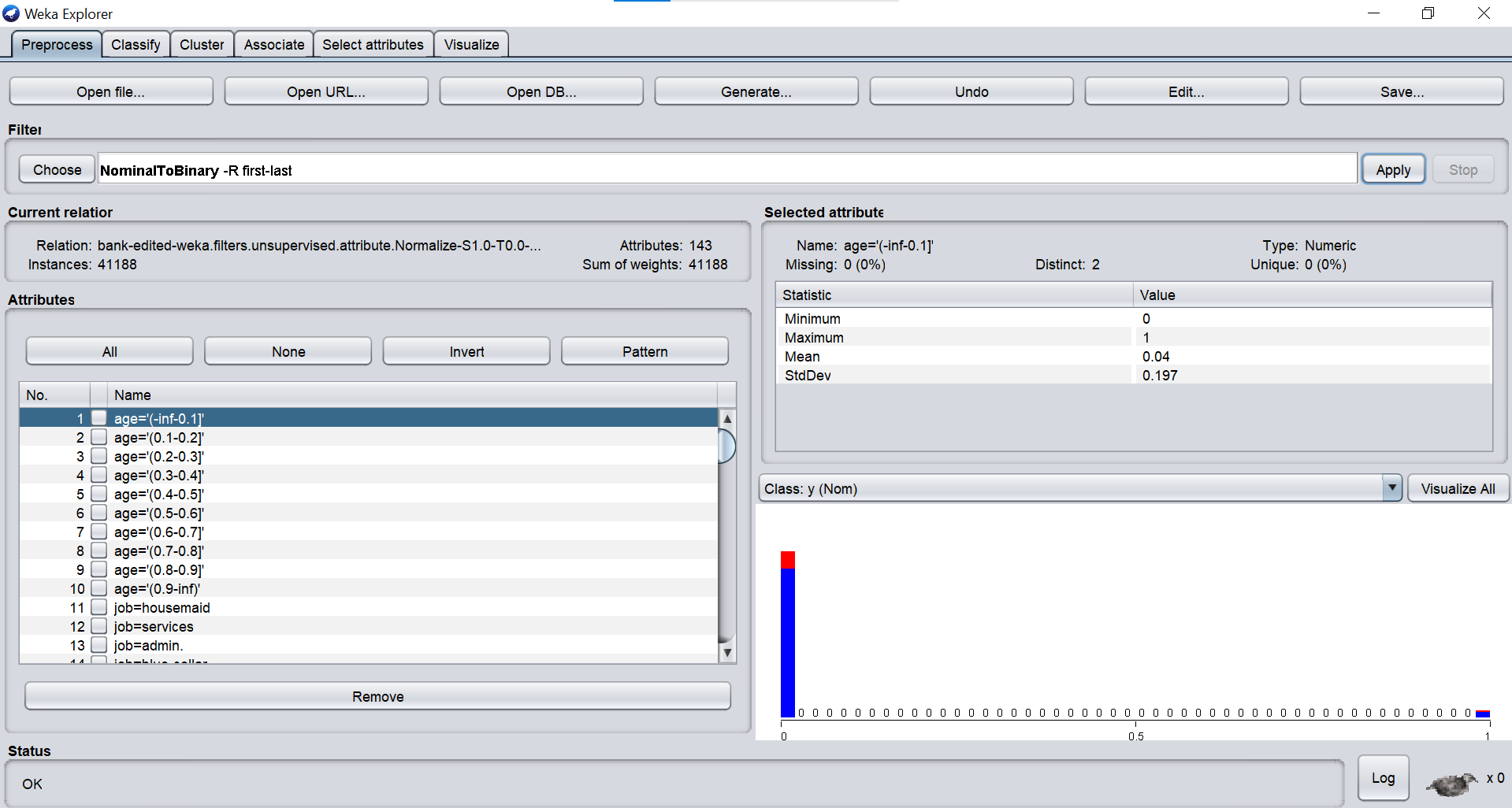
Hình 3-24: Chuyển đổi định dạng thành nhị phân

Chọn binaryAtributesNorimal =True



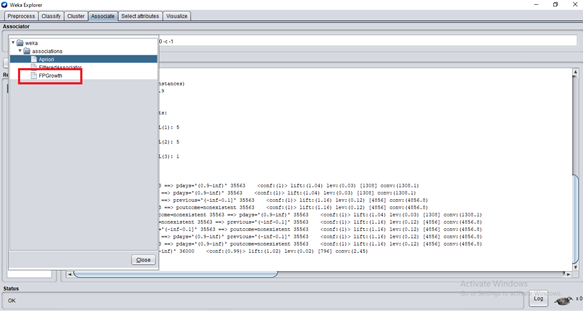
Hình 3-25: Chuyển đổi định dạng thành nhị phân

Chọn All - > chọn Apply



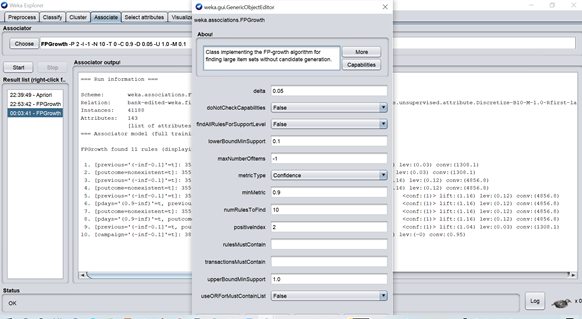
Hình 3-26: Chuyển đổi định dạng thành nhị phân

*Bước 5: Chọn luật kết hợp FB\_ Growth*

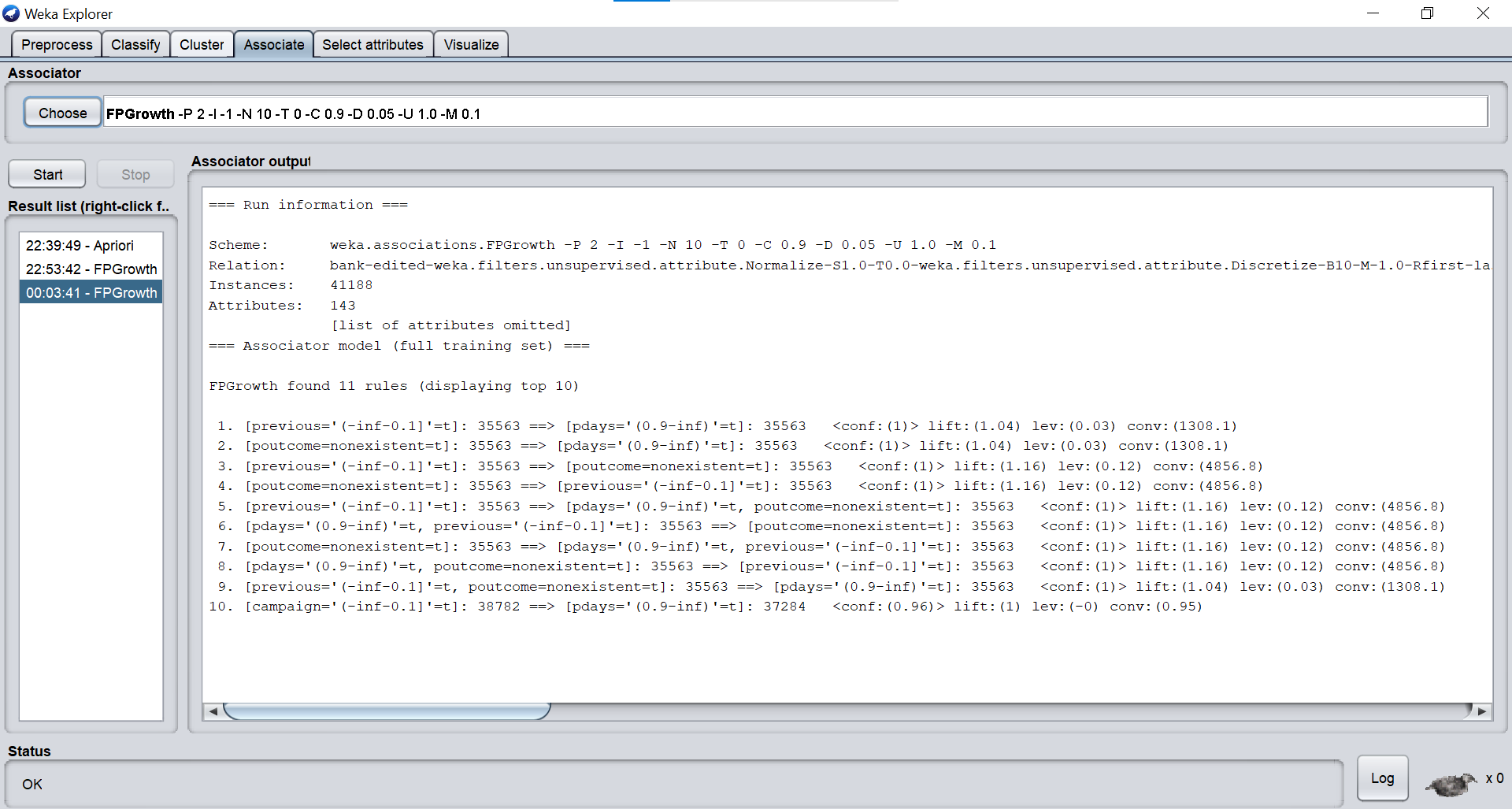


Hình 3-27: Chọn luật kết hợp FB\_Growth

*Bước 6: Chọn thông số*



Hình 3-28: Chọn thông số cho luật kết hợp FB\_Growth



Hình 3-27: Kết quả

Bước 7: Đọc kết quả

1. [previous='(-inf-0.1]'\_binarized=1]: 35563 ==> [pdays='(0.9-inf)'\_binarized=1]: 35563 <conf:(1)> lift:(1.04) lev:(0.03) conv:(1308.1)

=>> *Với thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1 ( 35563 trường hợp) =>> pday > 0.9 ( 35563 trường hợp) (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước với độ tin cậy là 100%.*

1. [poutcome=nonexistent\_binarized=1]: 35563 ==> [pdays='(0.9-inf)'\_binarized=1]: 35563 <conf:(1)> lift:(1.04) lev:(0.03) conv:(1308.1)

*=>> Với thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) =>> pday > 0.9 ( 35563 trường hợp) (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước với độ tin cậy là 100%.*

1. [previous='(-inf-0.1]'\_binarized=1]: 35563 ==> [poutcome=nonexistent\_binarized=1]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

*=>> Với thuộc tính previous ( số cuộc gọi được thực hiện) < 0,1 ( 35563 trường hợp) =>> poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) với độ tin cậy là 100%.*

1. [poutcome=nonexistent\_binarized=1]: 35563 ==> [previous='(-inf-0.1]'\_binarized=1]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

*=>>Với thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) =>>previous < 0,1 ( 35563 trường hợp)( số cuộc gọi được thực hiện với độ tin cậy là 100%.)*

1. [previous='(-inf-0.1]'\_binarized=1]: 35563 ==> [pdays='(0.9-inf)'\_binarized=1, poutcome=nonexistent\_binarized=1]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

=>> *Với thuộc tính previous ( số cuộc gọi được thực hiện) <= 0.1( 35563 trường hợp) =>>poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) và thuộc tính pday > 0.9 ( 35563 trường hợp) (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước với độ tin cậy là 100%.*

1. [pdays='(0.9-inf)'\_binarized=1, previous='(-inf-0.1]'\_binarized=1]: 35563 ==> [poutcome=nonexistent\_binarized=1]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

=>> *pday > 0.9 (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và thuộc tính* previous *( số cuộc gọi được thực hiện) < 0,1 ( 35563 trường hợp)=>> poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) với độ tin cậy là 100%.*

1. [poutcome=nonexistent\_binarized=1]: 35563 ==> [pdays='(0.9-inf)'\_binarized=1, previous='(-inf-0.1]'\_binarized=1]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

=>> *Với thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp) =>> pday > 0.9 (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và thuộc tính* previous *( số cuộc gọi được thực hiện) < 0,1 ( 35563 trường hợp) với độ tin cậy là 100%.*

1. [pdays='(0.9-inf)'\_binarized=1, poutcome=nonexistent\_binarized=1]: 35563 ==> [previous='(-inf-0.1]'\_binarized=1]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

=>> *pday > 0.9 (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước) và poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp)=>>* previous *( số cuộc gọi được thực hiện) < 0,1 ( 35563 trường hợp) với độ tin cậy là 100%.*

1. [previous='(-inf-0.1]'\_binarized=1, poutcome=nonexistent\_binarized=1]: 35563 ==> [pdays='(0.9-inf)'\_binarized=1]: 35563 <conf:(1)> lift:(1.04) lev:(0.03) conv:(1308.1)

*=>>Với thuộc tính previous ( số cuộc gọi được thực hiện) < 0.1( 35563 trường hợp) và thuộc tính poutcome ( kết quả của chiến dịch trước) = nonexistent ( 35563 trường hợp)=>>pday > 0.9 (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước)( 35563 trường hợp) với độ tin cậy là 100%.*

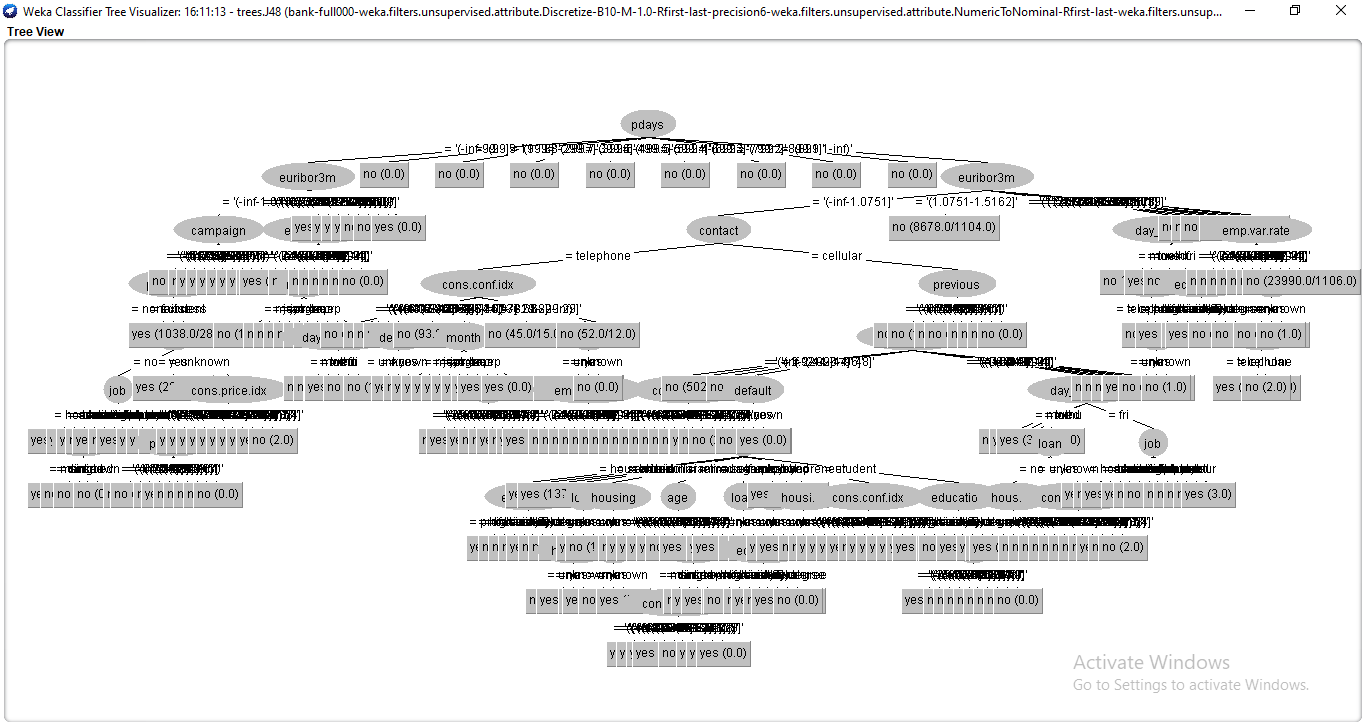
1. [campaign='(-inf-0.1]'\_binarized=1]: 38782 ==> [pdays='(0.9-inf)'\_binarized=1]: 37284 <conf:(0.96)> lift:(1) lev:(-0) conv:(0.95)

=>> Với thuộc tính campaign(Số lượng liên hệ được thực hiện trong chiến dịch này và cho khách hàng này) < 0,1  *( 38782 trường hợp) =>>pday > 0.9 (số ngày kể từ sau khi khách hàng được liên hệ lần cuối từ chiến dịch trước)( 35563 trường hợp) với độ tin cậy là 100%.*

1. **Biểu diễn dữ liệu**
   1. **Phân lớp dữ liệu**

Kết quả khi lựa chọn Percentage split để test với 55% dữ liệu dùng để test 45% còn lại dung để kiểm tra.

Số các bản ghi (đối tượng) được phân lớp đúng là 16658 chiếm gần 90% và số các bản ghi phân lớp sai là 1877 chiếm gần 10%.Tổng số đối tượng đưa vào phân lớp là 18535 bản ghi.



Hình 4-1: Kết quả cây phân lớp

Cây sinh ra có 304 lá và kích thước là 356

Kết quả cây như sau:

1.pdays = '(99.9-199.8]': no (0.0)

2.pdays = '(199.8-299.7]': no (0.0)

3.pdays = '(299.7-399.6]': no (0.0)

4.pdays = '(399.6-499.5]': no (0.0)

5.pdays = '(499.5-599.4]': no (0.0)

6.pdays = '(599.4-699.3]': no (0.0)

7.pdays = '(699.3-799.2]': no (0.0)

8.pdays = '(799.2-899.1]': no (0.0)

9.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = nonexistent: yes (0.0)

10.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = housemaid: yes (3.0/1.0)

11.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = services: yes (3.0)

12.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin -> marital = married: yes (14.0/4.0)

13.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin -> marital = single: no (13.0/2.0)

14.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin -> marital = divorced: no (2.0/1.0)

15.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no -> job = admin-> marital = unknown: no (0.0)

16.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = blue-collar: yes (6.0/2.0)

17.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = technician: no (15.0/5.0)

18.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = retired: yes (10.0/4.0)

19.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = management: no (8.0/2.0)

20.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = unemployed: yes (3.0/1.0)

21.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = self-employed: yes (2.0)

22.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = unknown: yes (1.0)

23.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = entrepreneur: no (1.0)

24.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(-inf-0.7]': no (0.0)

25.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(0.7-1.4]': no (0.0)

26.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(1.4-2.1]': no (10.0/5.0)

27.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(2.1-2.8]': no (0.0)

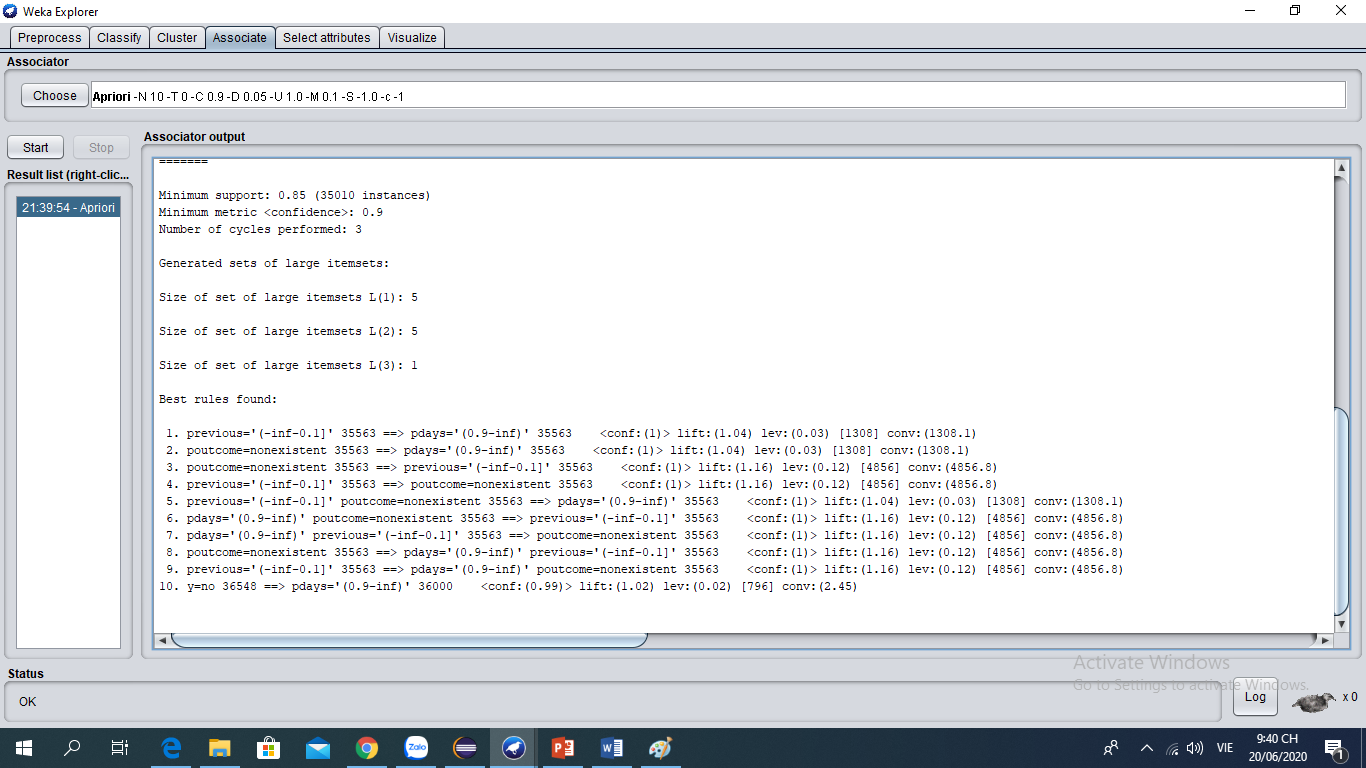
28.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(2.8-3.5]': yes (2.0)

29.pdays = '(-inf-99.9]' -> euribor3m = '(-inf-1.0751]' -> campaign = '(-inf-6.5]' -> poutcome = failure-> loan = no-> job = student-> previous = '(3.5-4.2]': no (4.0)

v.v…..

* 1. **Luật kết hợp**

Kết quả của thuật toán Apiori cho ta 10 luật kết hợp tốt nhất khi ta chọn minsupp =0.1 là:



Hình 4-2: Kết quả luật kết hợp Apiori

Kết quả sinh ra các luật sau:

1. previous='(-inf-0.1]' 35563 ==> pdays='(0.9-inf)' 35563 <conf:(1)> lift:(1.04) lev:(0.03) [1308] conv:(1308.1)

poutcome=nonexistent 35563 ==> pdays='(0.9-inf)' 35563 <conf:(1)> lift:(1.04) lev:(0.03) [1308] conv:(1308.1)

poutcome=nonexistent 35563 ==> previous='(-inf-0.1]' 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

previous='(-inf-0.1]' 35563 ==> poutcome=nonexistent 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

5. previous='(-inf-0.1]' poutcome=nonexistent 35563 ==> pdays='(0.9-inf)' 35563 <conf:(1)> lift:(1.04) lev:(0.03) [1308] conv:(1308.1)

6. pdays='(0.9-inf)' poutcome=nonexistent 35563 ==> previous='(-inf-0.1]' 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

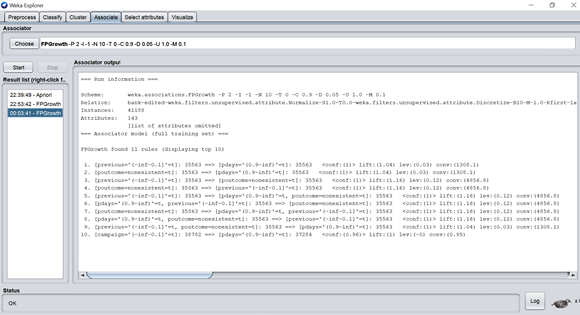
7. pdays='(0.9-inf)' previous='(-inf-0.1]' 35563 ==> poutcome=nonexistent 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

8. poutcome=nonexistent 35563 ==> pdays='(0.9-inf)' previous='(-inf-0.1]' 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

9. previous='(-inf-0.1]' 35563 ==> pdays='(0.9-inf)' poutcome=nonexistent 35563 <conf:(1)> lift:(1.16) lev:(0.12) [4856] conv:(4856.8)

1. y=no 36548 ==> pdays='(0.9-inf)' 36000 <conf:(0.99)> lift:(1.02) lev:(0.02) [796] conv:(2.45)

Kết quả của thuật toán FP growth cho ta 10 luật kết hợp tốt nhất khi ta chọn minsupp =0.1 là:



Hình 4-3: Kết quả luật kết hợp FP growth

Kết quả sinh ra các luật sau:

1. [previous='(-inf-0.1]'=t]: 35563 ==> [pdays='(0.9-inf)'=t]: 35563 <conf:(1)> lift:(1.04) lev:(0.03) conv:(1308.1)

2. [poutcome=nonexistent=t]: 35563 ==> [pdays='(0.9-inf)'=t]: 35563 <conf:(1)> lift:(1.04) lev:(0.03) conv:(1308.1)

3. [previous='(-inf-0.1]'=t]: 35563 ==> [poutcome=nonexistent=t]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

4. [poutcome=nonexistent=t]: 35563 ==> [previous='(-inf-0.1]'=t]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

5. [previous='(-inf-0.1]'=t]: 35563 ==> [pdays='(0.9-inf)'=t, poutcome=nonexistent=t]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

6. [pdays='(0.9-inf)'=t, previous='(-inf-0.1]'=t]: 35563 ==> [poutcome=nonexistent=t]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

7. [poutcome=nonexistent=t]: 35563 ==> [pdays='(0.9-inf)'=t, previous='(-inf-0.1]'=t]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

8. [pdays='(0.9-inf)'=t, poutcome=nonexistent=t]: 35563 ==> [previous='(-inf-0.1]'=t]: 35563 <conf:(1)> lift:(1.16) lev:(0.12) conv:(4856.8)

9. [previous='(-inf-0.1]'=t, poutcome=nonexistent=t]: 35563 ==> [pdays='(0.9-inf)'=t]: 35563 <conf:(1)> lift:(1.04) lev:(0.03) conv:(1308.1)

10. [campaign='(-inf-0.1]'=t]: 38782 ==> [pdays='(0.9-inf)'=t]: 37284 <conf:(0.96)> lift:(1) lev:(-0) conv:(0.95).

1. **Nhận xét đánh giá**
   1. **Nhận xét đánh giá**

Khai phá dữ liệu là một bước trong quy trình phát hiện tri thức. Hiện nay, mọi người không ngừng tìm tòi các kỹ thuật để thực hiện khai phá dữ liệu một cách nhanh nhất và có được kết quả tốt nhất.

Bộ môn: *Khai phá dữ liệu* là một môn học đầy bổ ích và hữu dụng cho sinh viên ngày nay. Trên thực tế, nhiều lĩnh vực như: ngân hàng, giáo dục, bệnh viện…đã áp dụng BI (Business Intelligence) vào để khai phá dữ liệu. Việc này giúp cho doanh nghiệp ngày càng lớn mạnh và phát triển hơn.

* 1. **Kết luận**

Trong bài tập lớn này, chúng em tìm hiểu và trình bày về hai kỹ thuật trong khai phá dữ liệu để phân lớp và luật kết hợp dữ liệu với đề tài “Khai phá dữ liệu trong lĩnh vực tiếp thị ngân hàng”. Mục đính của việc khai phá là phân loại khách hàng trong tương lai xem họ sẽ gửi một khoản tiền có kỳ hạn tại ngân hàng hay không. Để từ đó giúp cho việc tiếp thị ngân hàng dễ dàng hơn, chính xác hơn.

Trong quá trình làm bài tập lớn này, chúng em xin gửi lời cảm ơn đến thấy giáo Trần Mạnh Tuấn. Thầy đã rất tận tình hướng dẫn chi tiết cho chúng em, những kiến thức thầy cung cấp rất hữu ích .Chúng em rất mong nhận được những góp ý từ thầy.

* 1. **Tài liệu tham khảo**

[Tiền xử lý dữ liệu](https://techtalk.vn/blog/posts/tien-xu-ly-du-lieu-horse-colic-dataset)

[Phân lớp dữ liệu](http://uet.vnu.edu.vn/~thuyhq/Student_Thesis/K46_Nguyen_Thi_Thuy_Linh_Thesis.pdf)

[Lựa chọn thuộc tính và khái phá luật kết hợp](https://www.slideshare.net/HoQuangThanh/la-chn-thuc-tnh-v-khai-ph-lut-kt-hp-trn-weka)

[Weka Manual Book](https://drive.google.com/drive/folders/179_ZxKXvVakOsppUY-ZRkY5_p3HB5CCr?fbclid=IwAR3UDsQ85QehmPK2w5r0ajjlTUG3yVH0uBOlOD4-xSAWxkUQ2RP7abVp-WI)

**PHIẾU CHẤM ĐIỂM**

|  |  |  |
| --- | --- | --- |
| STT | Tên | Công việc |
| 1 | Đào Minh Khoa | * Tìm luật kết hợp cho bài toán * Chạy weka và viết báo cáo Chương 3 (Mục 3.2), Chương 4 (Mục 4.2) * Code sản phẩm |
| 2 | Nguyễn Quốc Hiệu | * Chạy weka và viết báo cáo Chương 3 (Mục 3.1), chương 4 (Mục 4.1). * Phân lớp dữ liệu * Code sản phẩm |
| 3 | Tô Văn Tuấn | * Viết Báo cáo Chương 1,2 * Tiền xử lý dữ liệu * Code sản phẩm |

**LỜI CẢM ƠN**

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè. Với lòng biết ơn sâu sắc nhất, chúng em xin gửi lời cảm ơn đến thầy Trần Mạnh Tuấn ở Khoa Công nghệ thông tin – Trường Đại Học Thủy Lợi với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Và đặc biệt, trong học kỳ này, khoa đã tổ chức cho chúng em được tiếp cận với môn học mà theo em là rất hữu ích đối với sinh viên ngành đặc biệt là trong sự phát triển của thời đại mới.

Trong quá trình làm bài tập lớn này thầy đã rất tận tình hướng dẫn chi tiết cho chúng em, những kiến thức thầy cung cấp rất hữu ích. Chúng em rất mong nhận được những góp ý từ thầy. Nhờ có những lời hướng dẫn, dạy bảo của thầy nên đề tài nghiên cứu của nhóm chúng em mới có thể hoàn thiện tốt đẹp.

Bước đầu đi vào thực tế của chúng em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp quý báu của Thầy để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình.

Nhóm em xin chân thành cảm ơn!