Trường đại học Khoa học Tự nhiên – đại học Quốc gia thành phố Hồ Chí Minh

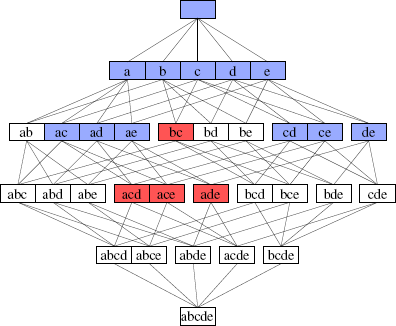
Khoa Công nghệ thông tin

---🙡🙣---

**Báo cáo thực hành**

Tập phổ biến và

khai thác luật kết hợp



Giáo viên hướng dẫn: **Nguyễn Ngọc Đức**

Lớp: **Khai thác dữ liệu và ứng dụng – CQ2017/21**

Nhóm 21: **Nguyễn Đăng Anh Thi 1612645**

**Nguyễn Huỳnh Xuân Mai 1712091**

**Âu Dương Tấn Sang 1712145**

**I. Tổng quan:**

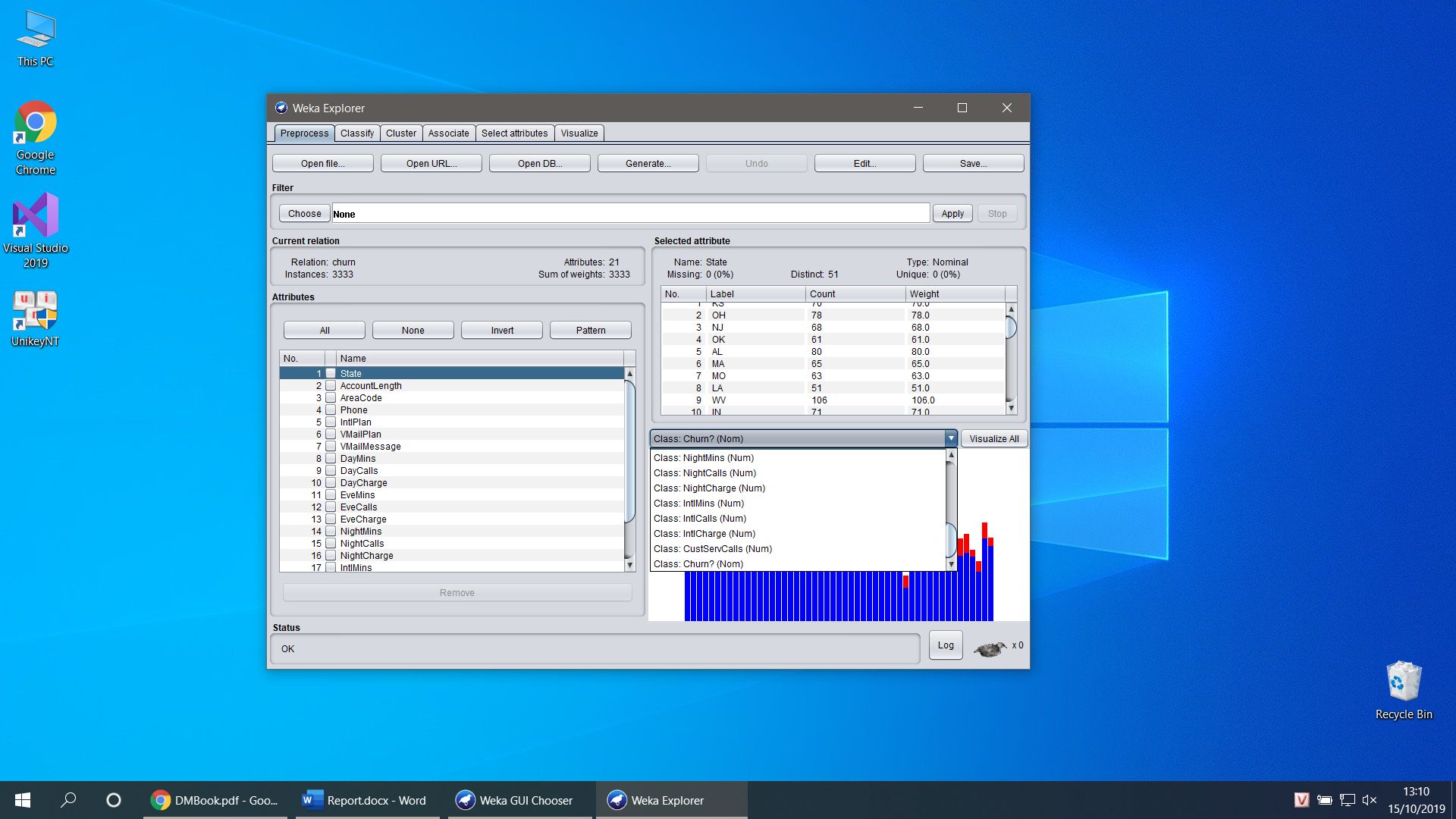
Khai thác luật kết hợp là một trong những phương thức hàng đầu để nhận dạng mẫu và khai thác tri thức từ các tập dữ liệu lớn. Mục tiêu của bài thực hành này là áp dụng các thuật toán khai thác luật kết hợp (được phần mềm Weka cung cấp) để giải quyết các vấn đề thực sự trong cuộc sống.

Tình huống đặt ra là một công ty viễn thông bỗng dưng có một lượng lớn khách hành ngưng sử dụng dịch vụ nhưng lại không rõ lí do vì sao. Bộ phận khai thác dữ liệu phải tìm hiểu vấn đề này thông qua việc khai thác luật kết hợp trên tập dữ liệu do công ty cung cấp ở đường dẫn dưới đây.

Dữ liệu: <http://ce.sharif.edu/courses/8586/1/ce925/assignments/files/assignDir2/churn.txt>

**II. Mô tả và tiền xử lý dữ liệu:**

Chuyển tập dữ liệu trên thành dạng .csv (hoặc .arff) và mở bằng Weka.



1. Mô tả dữ liệu

Tổng quan, tập dữ liệu có 21 thuộc tính và 3333 thể hiện. Dưới đây là tóm tắt sơ lược về tập dữ liệu này:

• State: **nominal**, mã bang

• AccountLength: **numeric**, thời gian hoạt động của khách hàng

• AreaCode: **numeric**, mã vùng

• Phone: **nominal**, số điện thoại

• IntlPlan: **nominal**, có hoặc không đăng kí InternationalCall

• VMailPlan: **nominal**, có hoặc không đăng kí VoiceMail

• VMailMessage: **numeric**, số lượng VoiceMail

• DayMins, EveMins, NightMins, IntlMins: **numeric**, thời gian gọi

• DayCalls, EveCalls, NightCalls, IntlCalls: **numeric**, số cuộc gọi

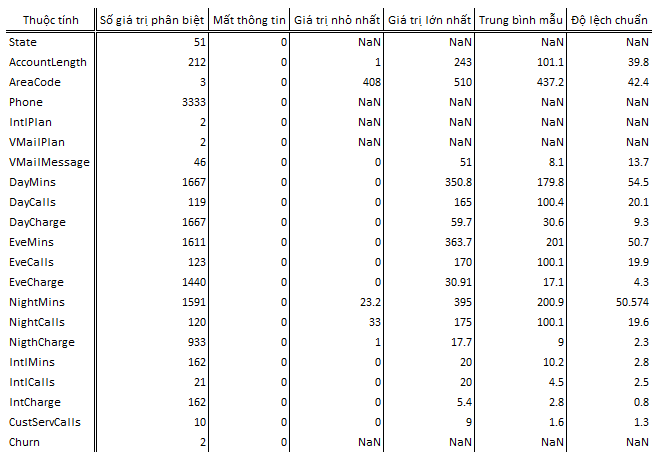
• DayCharge, EveCharge, NightCharge, IntlCharge: **numeric**, cước phí

• CustServCalls: **numeric**, số cuộc gọi đến CusomerService

• Churn: **nominal**, khách hàng có hoặc không ngưng sử dụng

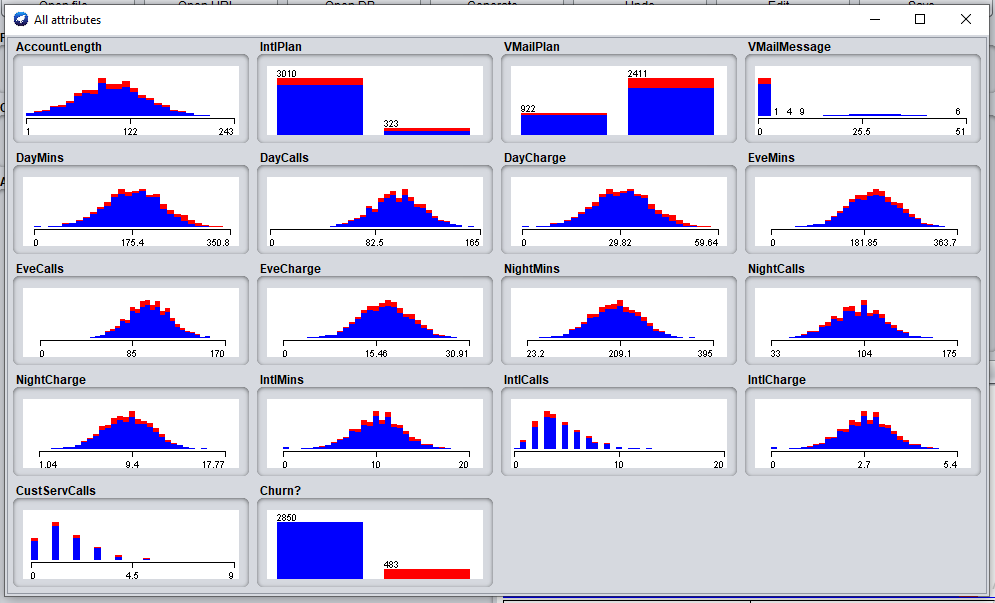
Các thuộc tính định danh như AreaCode, Phone, hay State có thể lược bỏ bớt do không thật sự ảnh hưởng đến quyết định của khách hàng.

Do không có giá trị nào bị khuyết nên có thể dễ dàng tính được trung bình mẫu, độ lệch chuẩn,… cho các thuộc tính numeric và và các độ đo thống kê khác cho các thuộc tính norminal. Trong đó, thuộc tính IntlPlan, VMailPlan và Churn là thuộc tính phân loại nhị phân (yes/no).



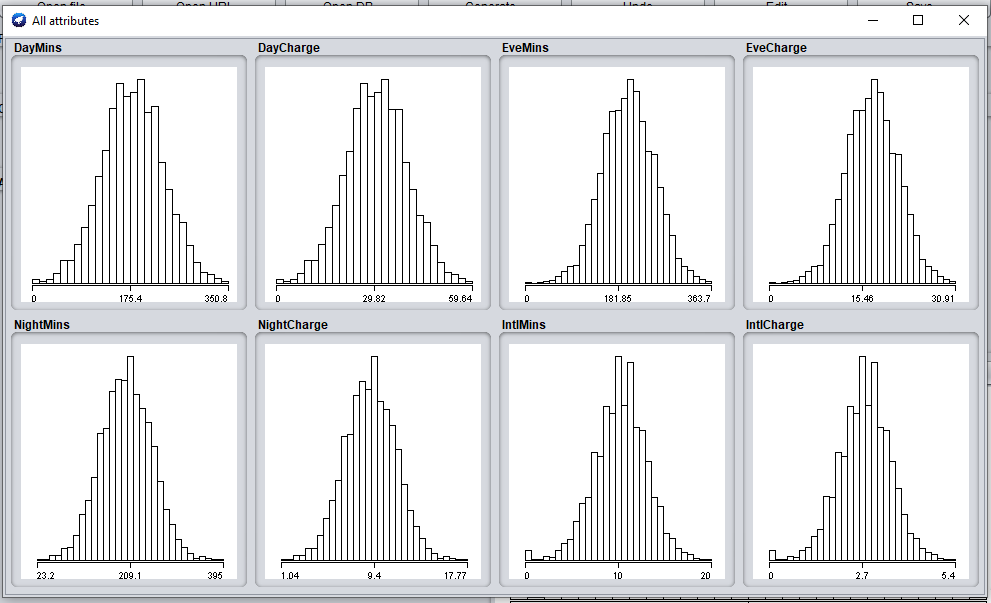
. Thống kê

Qua đó rút ra được nhận xét: các thuộc tính DayMins, EveMins và NightMins có phân bố khá rời rạc (độ lệch chuẩn rất cao), các thuộc tính VmailMessage, IntlCharge và CustServCalls bị lệch so với phân phối chuẩn.



. Phân phối dữ liệu

Hơn nữa, các thuộc tính Charge gần như thay đổi tuyến tính so với Mins, nên có thể lược bớt đi một trong 2 loại thuộc tính này. Ở đây, ta sẽ bỏ đi các thuộc tính Charge.



. Mins vs Charge

Tiếp theo tiến hành lọc bớt các thể hiện dữ liệu ngoại lai (outliers). Có nhiều phương pháp khác nhau cho thao tác này. Nhóm đã sử dụng module chuẩn hóa Z đã cài đặt ở lab01 để xác định các ngoại lai này.

Đặt file churn.csv và preprocess.py vào cùng thư mục, sau đó mở command line tại thư mục này và nhập lệnh sau:

python3 preprocess.py --input churn.txt --ouput normalized.csv --task zScoreNorm --prop prop1 prop2 …

Trong đó prop1, prop2,… là các thuộc tính muốn chuẩn hóa. Sau đó chọn ngưỡng threshold để loại bỏ các ngoại lai, giữ lại các thể hiện có giá trị nằm trong khoảng [-threshold, threshold]:

python3 preprocess.py --input normalized.csv --output removed.csv --task removeOutlier --threshold thresholdValue --prop prop1 prop2 …

**II. Khai thác luật kết hợp:**

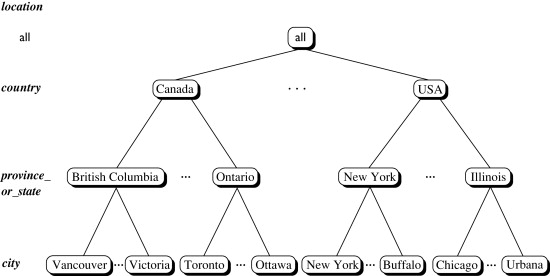
Với mỗi nhu cầu khác nhau, cách nhìn dữ liệu cũng khác nhau. Vì vậy ta có khái niệm phân cấp dữ liệu. Có nhiều cách phân cấp cho một thuộc tính dữ liệu nào đó, chẳng hạn:

• Địa chỉ: quốc gia > bang > thành phố > …

• Tuổi: thanh niên < trung niên < lão niên < …

• …

Việc phân cấp như vậy sẽ chia tập dữ liệu thành nhiều nhóm, mỗi nhóm có một ngữ nghĩa, vai trò hay độ ưu tiên khác nhau. Điều này giúp các thuật toán dễ dàng tìm thấy các luật kết hợp giàu ý nghĩa hơn.



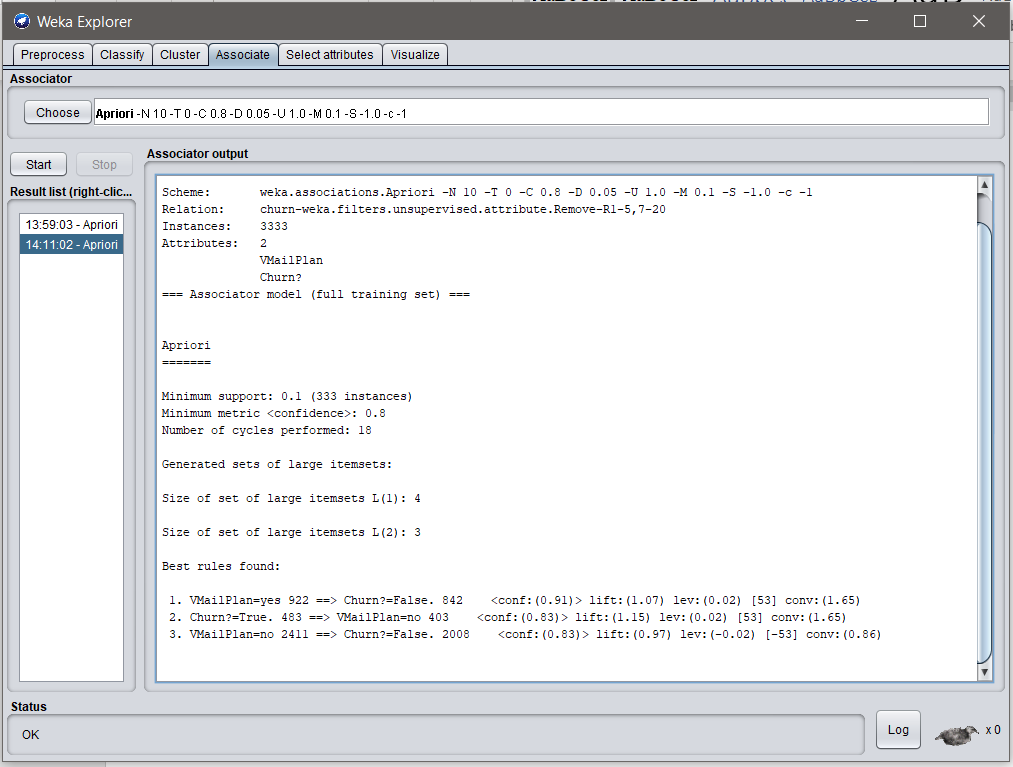
. Phân cấp dữ liệu

Dưới đây là một số thử nghiệm mà nhóm đã thực hiện.

**a. Khai thác luật kết hợp trên các thuộc tính nominal:**

Tập dữ liệu có hai thuộc tính nominal đáng quan tâm, đó là IntlPlan và VMailPlan, tương ứng với có/không đăng kí dịch vụ InternationalCall và VoiceMail. Khai thác luật kết hợp từ hai thuộc tính này có thể cho biết dịch vụ nào đang gặp vấn đền, dịch vụ nào cần kích cầu tiêu dung của khách hàng.

Thực hiện lọc ra thuộc tính VMailPlan và Churn, mở tab Associate và chọn thuật toán Apriori, nhập lowerBoundMinSupport là 0.1, minMetric (metric là Confidence) là 0.8, rồi bấm Start. Thu được kết quả như sau:



. VMailPlan

Kết quả trên cho thấy, Apriori khai thác được 3 luật với minSup = 0.1 và minCon = 0.8:

• VMailPlan = yes => Churn = False <conf: 0.91> <lift: 1.07> <conv: 1.65>

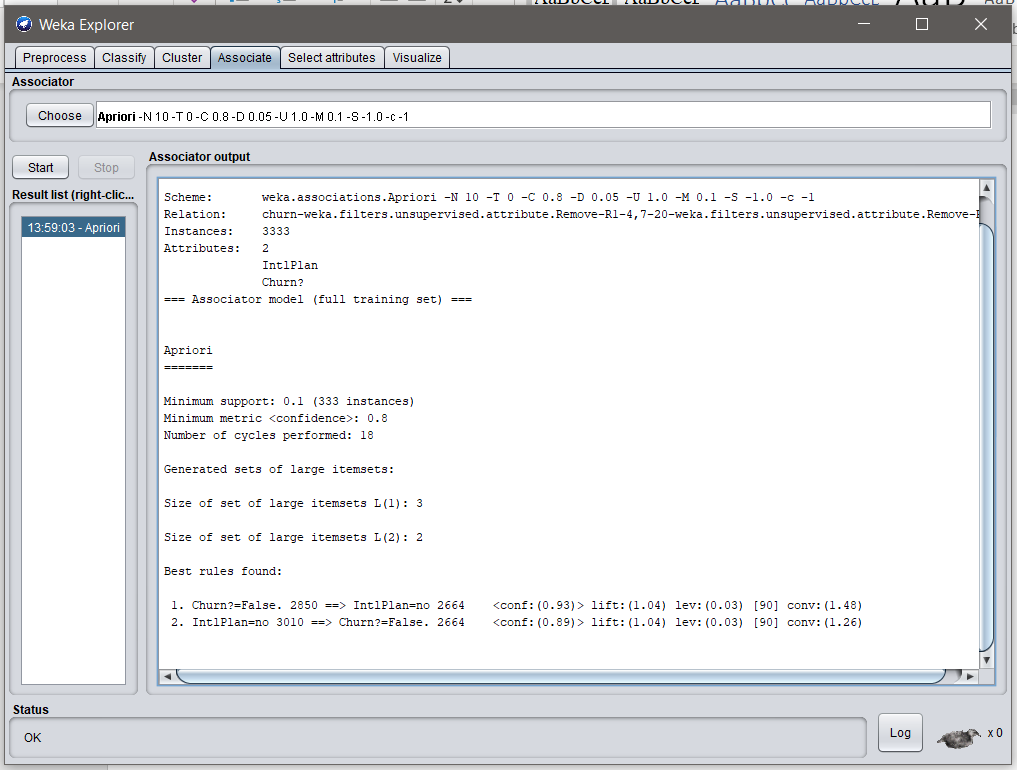
• Churn = True => VMailPlan = no <conf: 0.83> <lift: 1.15> <conv: 1.65>

• VMailPlan = no => Churn = False <conf: 0.83> <lift: 0.97> <conv: 0.86>

Có nghĩa rằng: 91% khách hàng đăng kí dịch vụ VoiceMall tiếp tục sử dụng, những khách hàng không sử dụng nữa thì 83% là do không đăng kí VoiceMail. Có thể bỏ qua luật cuối cùng do lift và conv không cao (bé hơn 1).

***Vậy, những khách hàng không đăng kí VoiceMail có khả năng ngưng sử dụng cao hơn những khách hàng có đăng kí. Công ty cần khuyến khích khách hàng sử dụng dịch vụ này.***

Thực hiện tương tự trên thuộc tính IntlPlan, thu được kết quả như sau:



. IntlPlan

Thuật toán Apriori thu được 2 luật kết hợp:

• Churn = False => IntlPlan = no <conf: 0.93> <lift: 1.04> <conv: 1.48>

• IntlPlan = no => Churn = False <conf: 0.89> <lift: 1.04> <conv: 1.26>

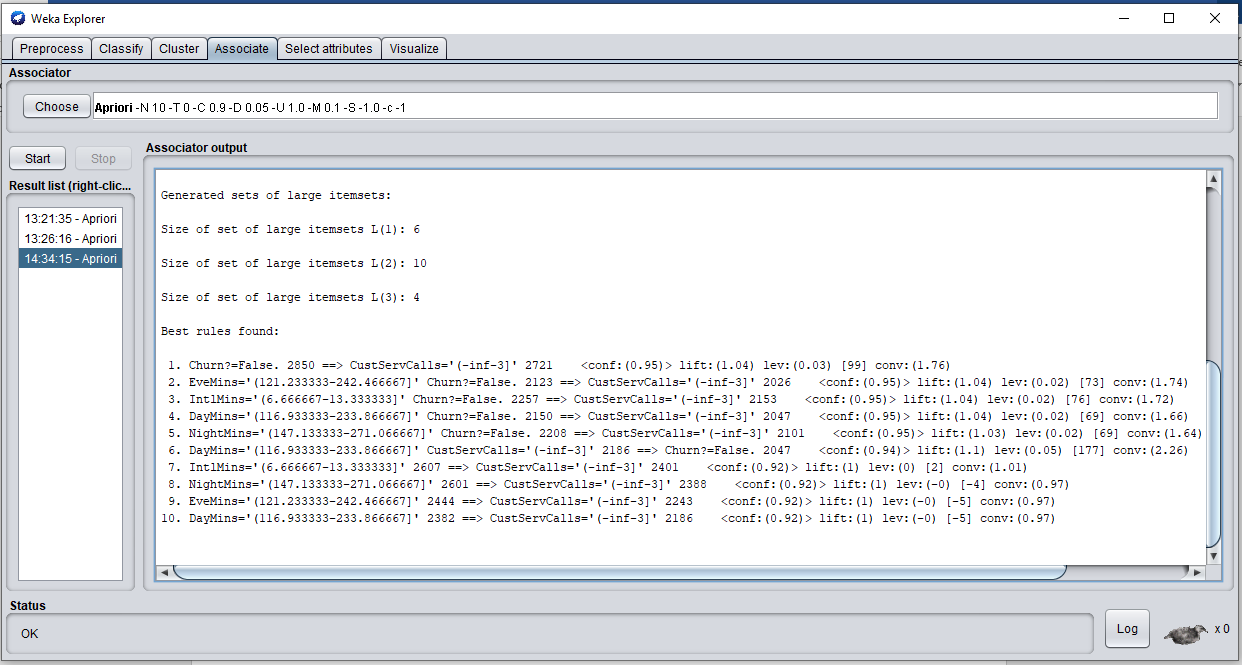
Nghĩa là: 93% khách hàng đang sử dụng là không đăng kí InternationalCall và 89% khách hàng không đăng kí dịch vụ InternationalCall vẫn đang sử dụng.

***Dịch vụ InternationalCall dường như có chút vấn đề khi phần lớn khách hàng đang hoạt động là không đăng kí dịch vụ này (93%).***

**b. Khai thác luật kết hợp trên các thuộc tính numeric:**

Các thuộc tính numeric tương đối khó khăn để khai thác luật kết hợp. Cụ thể là khó khăn trong việc rời rạc hóa dữ liệu vào những phân cấp như đã đề cập ở trên để có thể thu được các luật kết hợp có nhiều ý nghĩa nhất.

Chọn ra các thuộc tính DayMins, EveMins, NightMins, IntlMins, CustServCalls và Churn. Tiếp theo, rời rạc hóa bằng cách chia mỗi thuộc tính vào 3 giỏ với độ rộng bằng nhau (equal-width binning) với ý nghĩa: ít < trung bình < nhiều. Sau đó mở tab Associate, chọn thuật toán Apriori, nhập lowerBoundMinSupport là 0.1, minMetric (metric là Confidence) là 0.9, rồi bấm Start. Thu được kết quả như sau:



8. Thuộc tính numeric đã rởi rạc hóa

Apriori tìm được khá nhiều luật, nhưng hãy để ý đến luật thứ 6 vì luật này có chỉ số conv khá cao:

• DayMins = '(116.933333-233.866667]' CustServCalls='(-inf-3]' => Churn=False <conf: 0.94> <lift: 1.1> <conv: 2.26>

Luật này cho biết 94% các khách hàng có số phút gọi ban ngày ở nhóm trung bình và ít khi gọi tới dịch vụ chăm sóc khách hàng thì sẽ tiếp tục sử dụng.