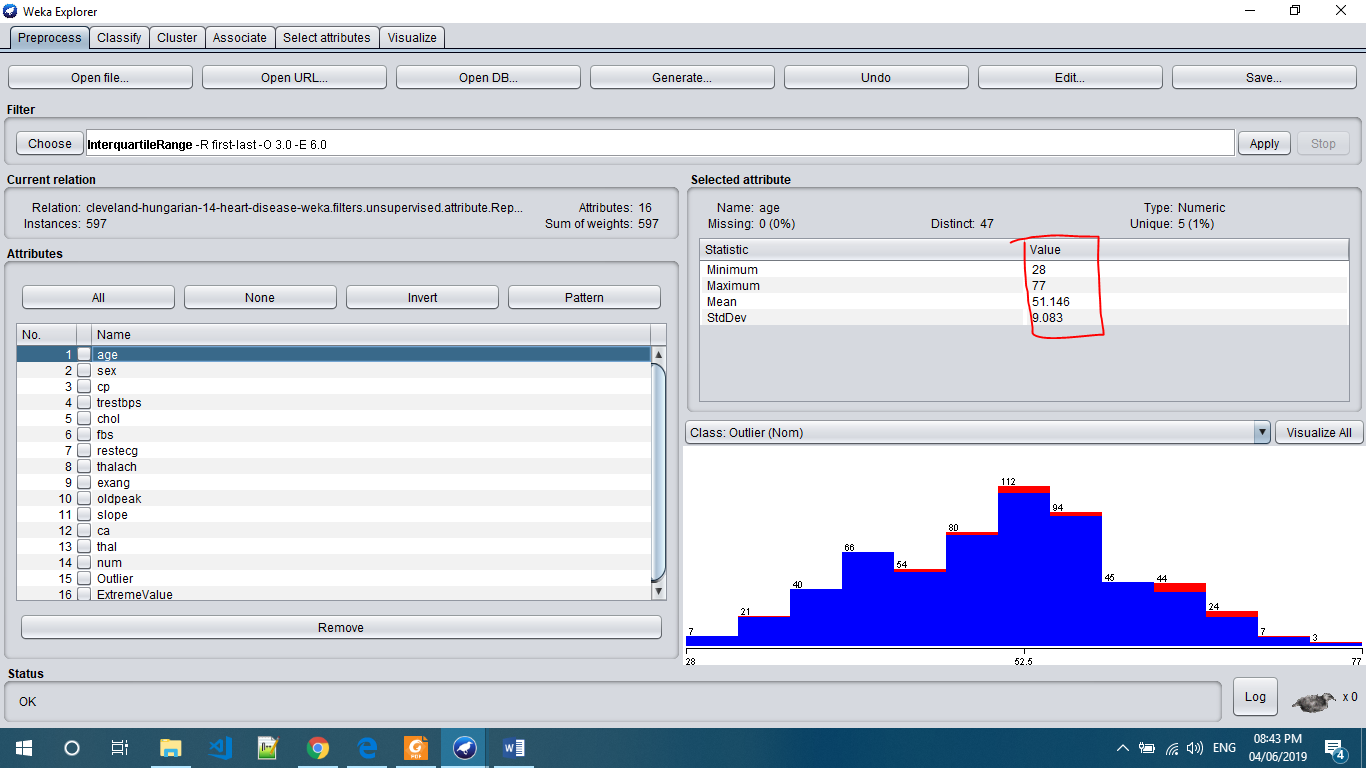
1. integration
2. Định nghĩa: sự tích hợp dữ liệu là sự hợp nhất dữ liệu từ nhiều nguồn dữ liệu, kết hợp dữ liệu từ nhiều nguồn vào kho dữ liệu lưu trữ mạch lạc.
3. Không, vì 2 dataset này có các id thuộc tính tương tự nhau.
4. Có dư thừa (redundancy) dữ liệu hay không??
5. Có mâu thuẫn dữ liệu (data value conflict) hay không?
7. Descriptive data summarization
8. Xét thuộc tính ‘age’:

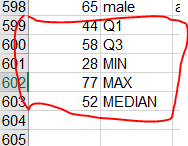
* Trung bình: mean = 51.146.
* Độ lệch chuẩn: sd = 9.038.
* Giá trị lớn nhất: maximum = 77.
* Giá trị nhỏ nhất: minimum = 28.



(2.1)

1. Five-number summary của thuộc tính age

* Minimum = 28.
* Q1 = 44.
* Median = 52.
* Q3 = 58.
* Maximum = 77.



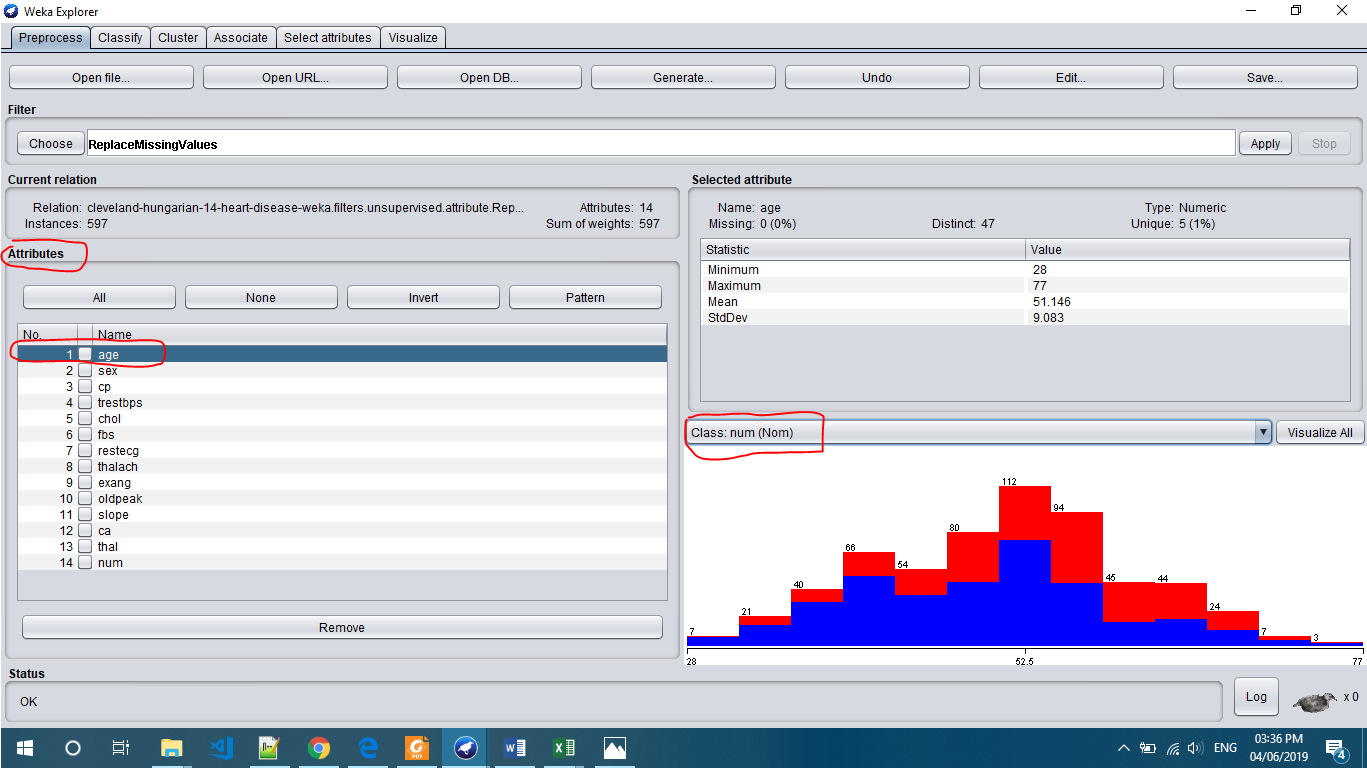
(2.2)

Weka chỉ cung cấp minimum và maximum. Q1, Q3 và median được nhóm chuyển file arff sang csv sao đó sử dụng excel để tìm các giá trị Q1, Q3 và median.

1. Thông tin thuộc tính

* Thuộc tính số (numeric): age, trestbps, chol, thalach, oldspeak, ca (6 thuộc tính).
* Thuộc tính có thứ tự (ordernal): restecg, slope, thal, num (4 thuộc tính).
* Thuộc tính rời rạc/danh sách (categorical/nominal): sex, cp, fbs, exang (4 thuộc tính).

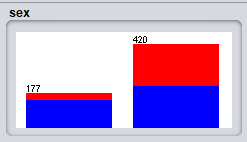
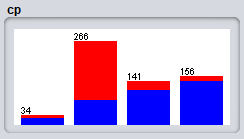
1. - Đồ thị này là đồ thị để thể hiện sự phân bố của thuộc tính hiện được chọn (nền xanh) ở khung Attributes và được phân lớp (classification) theo thuộc tính được chọn ở box Class.



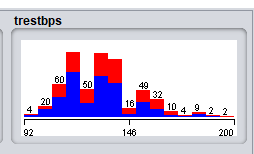
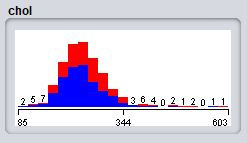
(2.3)

* Tên đồ thị: Đồ thị phân bố của thuộc tính.
* Màu xanh ứng với num có label ‘<50’ (không mắc bệnh). Màu đỏ ứng với num có label ‘>50\_1’ (măc bệnh mức 1).(do mức 2, 3, 4 không có nên trên đồ thị không hiển thị).
* Mỗi cột biểu diễn một khoảng tuổi (đang xét theo thuộc tính tuổi), giá trị trên mỗi cột là số lượng instances có ‘age’ thuộc khoảng đó.
* Đồ thị biểu diễn phân bố tuổi và tỉ lệ giữa người có bệnh và không có bệnh trên mỗi khoảng tuổi được chia. Ví dụ, ở khoảng [28, 31.769] có 7 người tỉ lệ người không mắc bệnh trong độ tuổi này chiếm cao hơn tỉ lệ người mắc bệnh mức 1 (màu xanh chiếm gần hết cột).

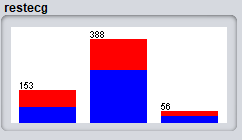
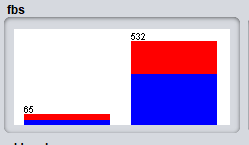


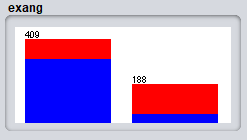
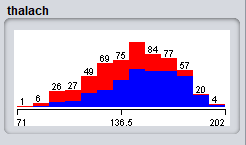
(Đồ thị phân bố của sex (2.4) và đồ thị phân bố của cp (chest-pain) (2.5) )

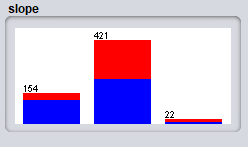
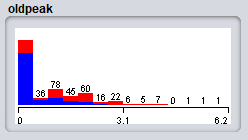
(Đồ thị phân bố của trestbps (2.6) và đồ thị phân bố của chol (2.7))



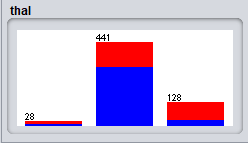
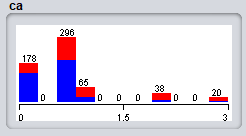
(Đồ thị phân bố của fbs (2.8) và đồ thị phân bố của restecg (2.9))



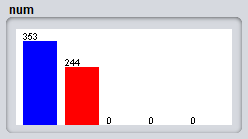
(Đồ thị phân bố của thalach (2.10) và đồ thị phân bố của exang (2.11))



(Đồ thị phân bố của oldspeak (2.12) và đồ thị phân bố của slope (2.13))



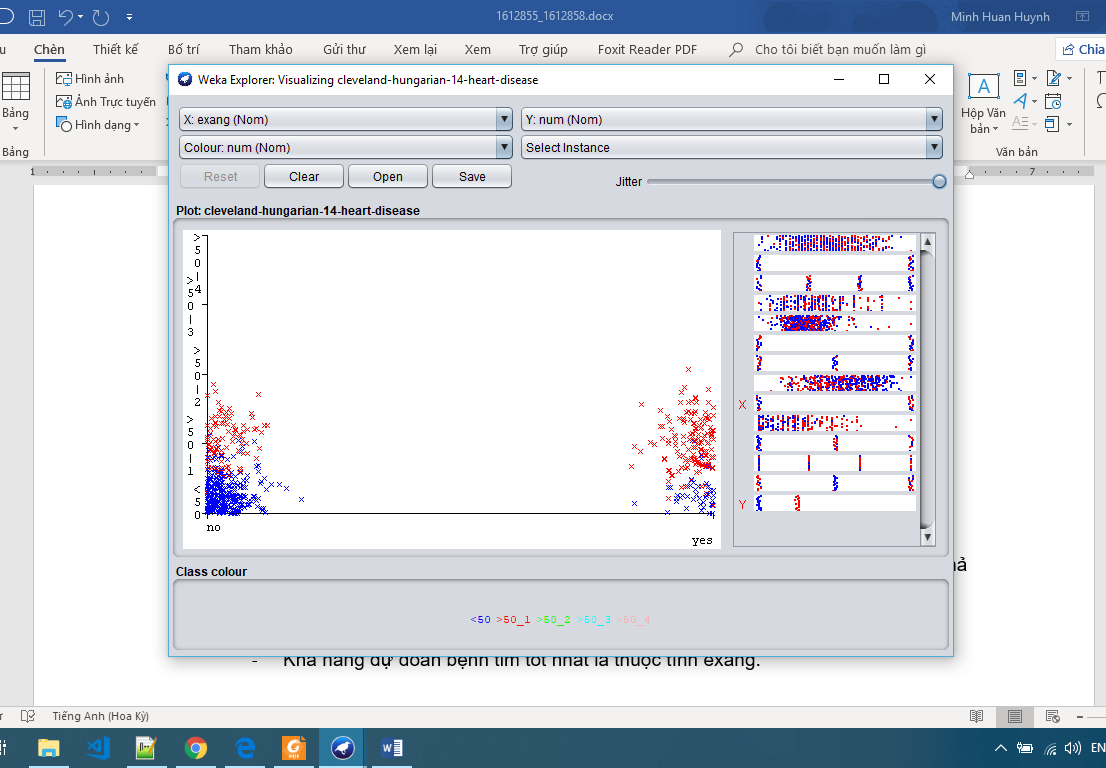
(Đồ thị phân bố của ca (2.14) và đồ thị phân bố của thal (2.15))



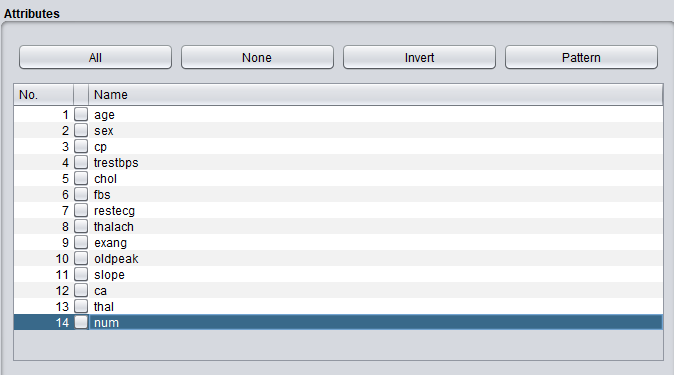
(Đồ thị phân bố của num (2.16))

1. Nhận xét:
2. Thuật ngữ sử dụng trong textbook để đặt tên cho đồ thị ở mục Visualize là “scatter plot”.

* Theo nhóm, các thuộc tính có vẽ dẫn đến bệnh tim là: oldspeak (không mắc bệnh trong khoản 0 – 1.6, lớn hơn 1.6 khả năng mắc bệnh cao), exang (no – khả năng không mắc bệnh cao, yes – khả năng mắc bệnh cao), thalach (<136.5 – khả năng mắc bệnh cao, >136.5 – thường không mắc bệnh), slope (up – không mắc bệnh cao, flat – khả năng mắc bệnh cao).
* Khả năng dự đoán bệnh tim tốt nhất là thuộc tính exang.



1. (Hoài làm chỗ này cái tương quan nha)
2. Selection
3. Có 14 thuộc tính trong datasets trước khi xử lý dữ liệu.



(3.1)

1. Các lựa chọn khác nhau để lựa chọn thuộc tính là:

Các phương pháp đánh giá:

* CfsSubsetEval: Đánh giá giá trị của một tập hợp con các thuộc tính bằng cách xem xét khả năng dự đoán riêng của từng tính năng cùng với mức độ dư thừa giữa chúng.
* ClassifierSubsetEval: Đánh giá các tập hợp thuộc tính trên dữ liệu huấn luyện hoặc một bộ kiểm tra riêng biệt.
* CorrelationAttributeEval: Đánh giá giá trị của một thuộc tính bằng cách đo lường mối tương quan (Pearson's) giữa nó và lớp. Các thuộc tính danh nghĩa được xem xét trên một giá trị theo cơ sở giá trị bằng cách coi mỗi giá trị là một chỉ báo.
* GainRatioAttributeEval: Đánh giá theo độ đo tỉ lệ đạt được với các lớp. với công thức tính gain là GainR(Class, Attribute) = (H(Class) - H(Class | Attribute)) / H(Attribute).
* OneRAttributeEval: Đánh giá giá trị của một thuộc tính bằng cách sử dụng trình phân loại OneR.
* PrincipalComponents: Thực hiện phân tích thành phần chính (chọn ra các thành phần chính) nhằm biến đổi dữ liệu (transformation data).
* ReliefFAttributeEval: Đánh giá giá trị của một thuộc tính bằng cách lặp lại việc lấy mẫu (sampling) của một instance và xem xét giá trị của thuộc tính đã cho với instance gần nhất của cùng một lớp và khác lớp.
* SymmetricalUncertAttributeEval: Đánh giá giá trị của một thuộc tính bằng cách đo độ không đảm bảo đối xứng đối với lớp.
* WrapperSubsetEval: đánh giá bằng tập bao các phân loại (“wrapper” method wraps a classifier in cross-validation loop – nguồn <https://www.futurelearn.com/courses/more-data-mining-with-weka/0/steps/29132>).

Các phương pháp tìm kiếm (chọn thuộc tính theo nhu cầu):

* BestFirst: Tìm kiếm tập con không gian các thuộc tính bằng tham lam tăng cường với cơ sở quay lui.
* GreedyStepwise: Phương pháp tìm kiếm tham lam tiến hay lùi thông qua tập con không gian các thuộc tính.
* Ranker: xếp hạng các thuộc tính theo giá trị nó được đánh giá.

1. So sánh các phương pháp giữa weka và textbook: trong textbook không thấy phương pháp tìm kiếm (chọn thuộc tính theo yêu cầu) là BestFirst và Ranker
2. Cleaning
3. Missing values (dữ liệu thiếu):

* Các phương pháp xử lý dữ liệu thiếu:

1. Bỏ qua bộ đó (ignore the tuple).
2. Điền các dữ liệu thiếu bằng phương pháp thủ công (mannually).
3. Sử dụng giá trị hằng ngoại vi (global constant) để điền: “Unknown” hay “-∞”.
4. Dùng thuộc tính trung bình (mean) để điền các giá trị thiếu.
5. Sử dụng thuộc tính trung bình cho tất cả các mẫu thuộc cùng một lớp với bộ dữ liệu đã cho.
6. Dùng giá trị có thể xảy ra nhất để điền vào giá trị còn thiếu.

* Weka đã cài đặt phương pháp: thay thế dữ liệu bằng giá trị trung bình hay mode, thay thế giá trị thiếu bằng hằng số do người dùng đặt, thay thế giá trị thiếu bằng giá trị có thể xảy ra nhất.
* Chọn phương pháp điền giá trị thiếu bằng mean hay mode. Vì

1. không thể tiến hành điền mannual vì dữ liệu thiếu nhiều ở thuộc tính ca (50%), thal (45%), slope (32%) và không thể biết được nên điền như thế nào.
2. Nếu chọn lược bỏ các tuple thiếu thì lại làm mất tính khách quan dữ liệu.
3. Chỉ có 1 số thuộc tính bị thiếu nên nhóm nghĩ tốt nhất là chọn ReplaceMissingValue by mean or mode sẽ đảm bảo.

* Cài đặt phương pháp

1. Noisy data (dữ liệu nhiễu):

* Các phương pháp loại bỏ dữ liệu nhiễu:

1. Phương pháp phân khoảng hay chia giỏ (bining): Chia giỏ theo giá trị trung bình (bin mean), chia giỏ theo trung vị (bin median), chia giỏ theo biên (bin boundaries).
2. Hồi quy (regression)
3. Phân cụm, gom nhóm (clustering)

* Weka hỗ trợ:

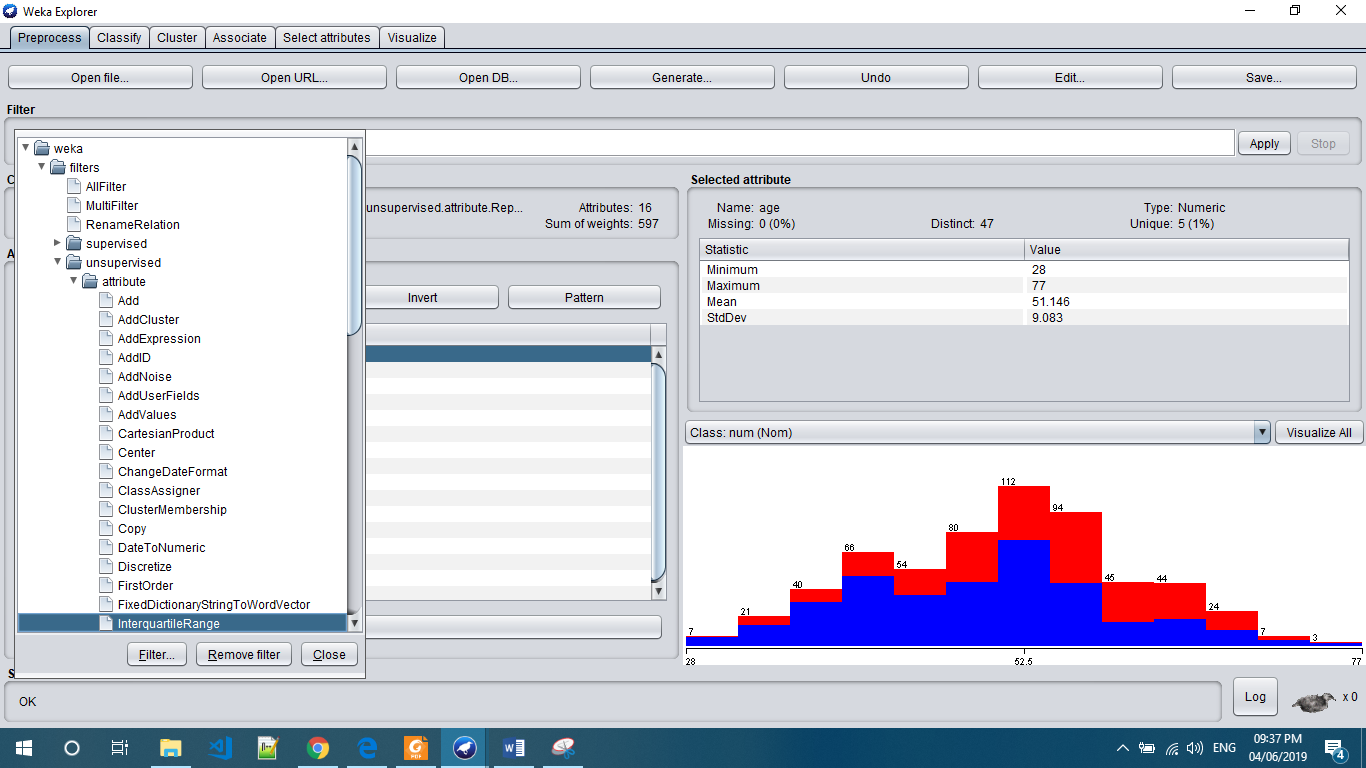
1. Hồi quy
2. Outlier data (dữ liệu ngoại lệ/ dữ liệu tạp):

* Các phương pháp:

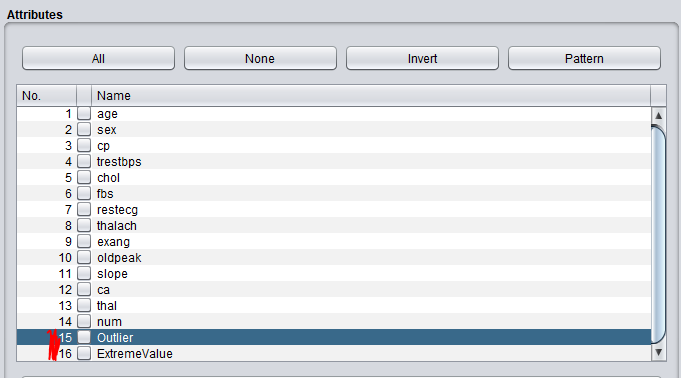
1. Gom cụm (Clustering)
2. Numeric outlier (IQR): điểm dữ liệu tạp nằm ngoài khoảng interquartile (IQR).
3. Z-score

* Dò tìm dữ liệu tạp bằng Weka:

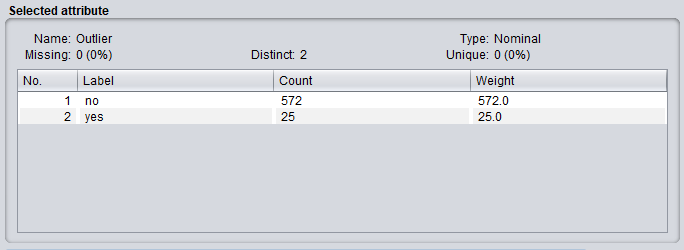
1. B1: Mục Filter -> Choose -> filters -> unsupervised -> attribute -> InterquartileRange.



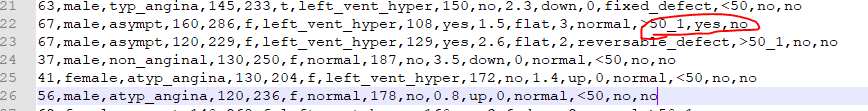
1. B2: Nhấn apply, sẽ xuất hiện thêm 2 thuộc tính trong mục Attributes là Outlier và ExtremeValue

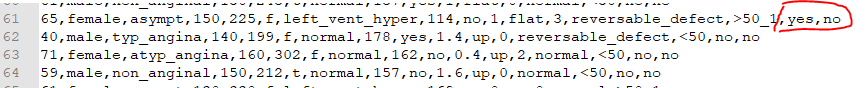


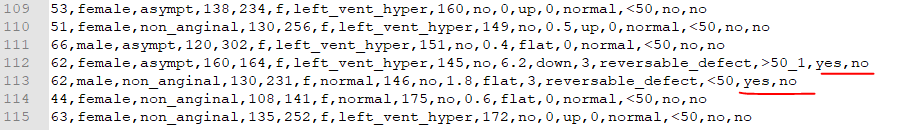
1. B3: Chọn thuộc tính Outlier để xem kết quả, có dữ liệu tạp hay không.



* Có dữ liệu tạp trong dataset (như hình trên) là 25 (thuộc tính gần cuối là thuộc tính outlier, file được lưu lại sau khi thực hiện các bước trên).

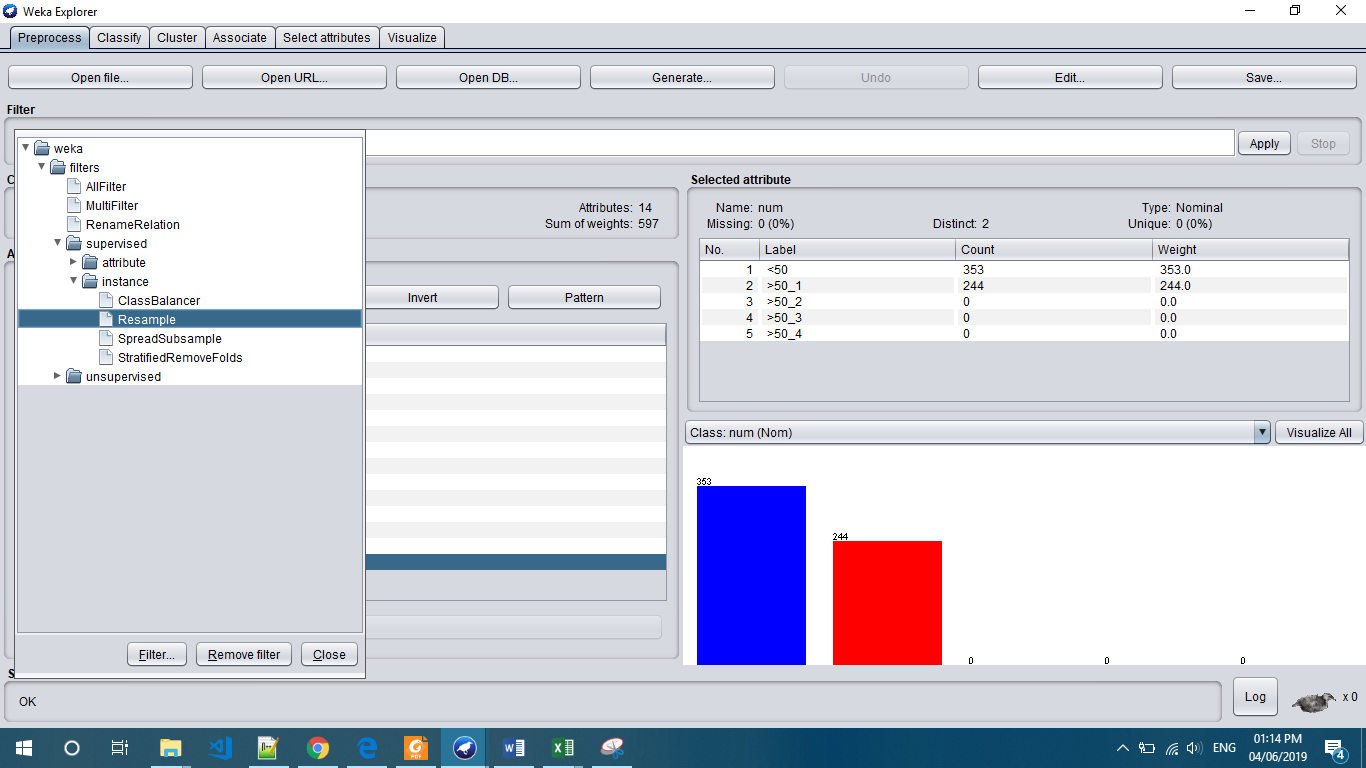






1. Lưu heart-cleaned.arff
2. Transformation
3. Reduction
4. Làm cách nào để lấy mẫu dữ liệu với các bộ lọc Weka:

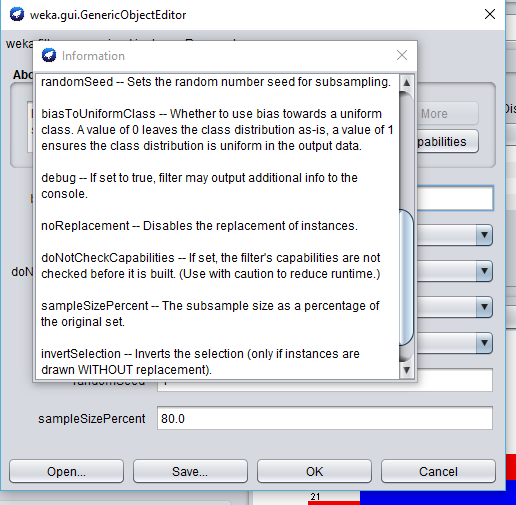
* B1: Ở mục Filter chọn nút Choose -> Chọn mục supervised (hoặc unsupervised) -> instance -> Resample.



(6.1)

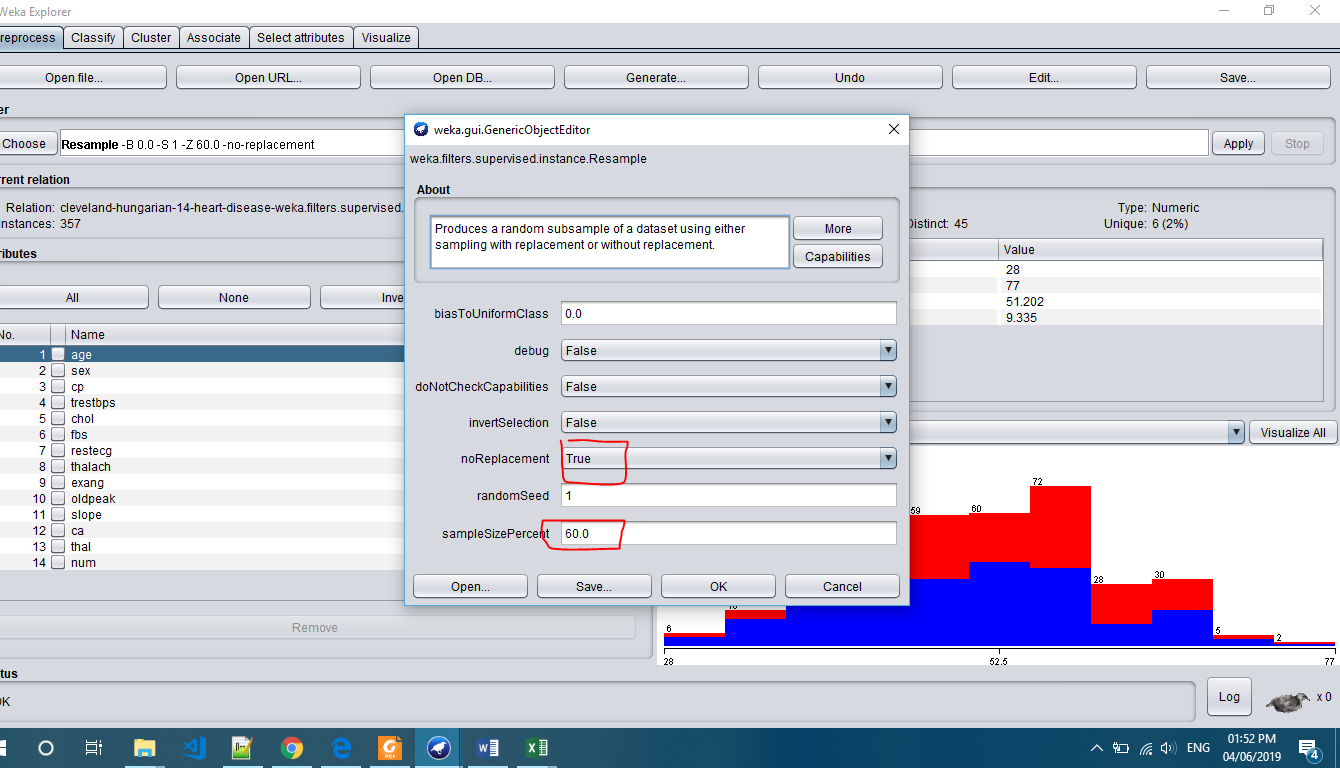
* B2: nhấp chọn ô thông số bên cạnh nút Choose để chỉnh sửa các thông số:

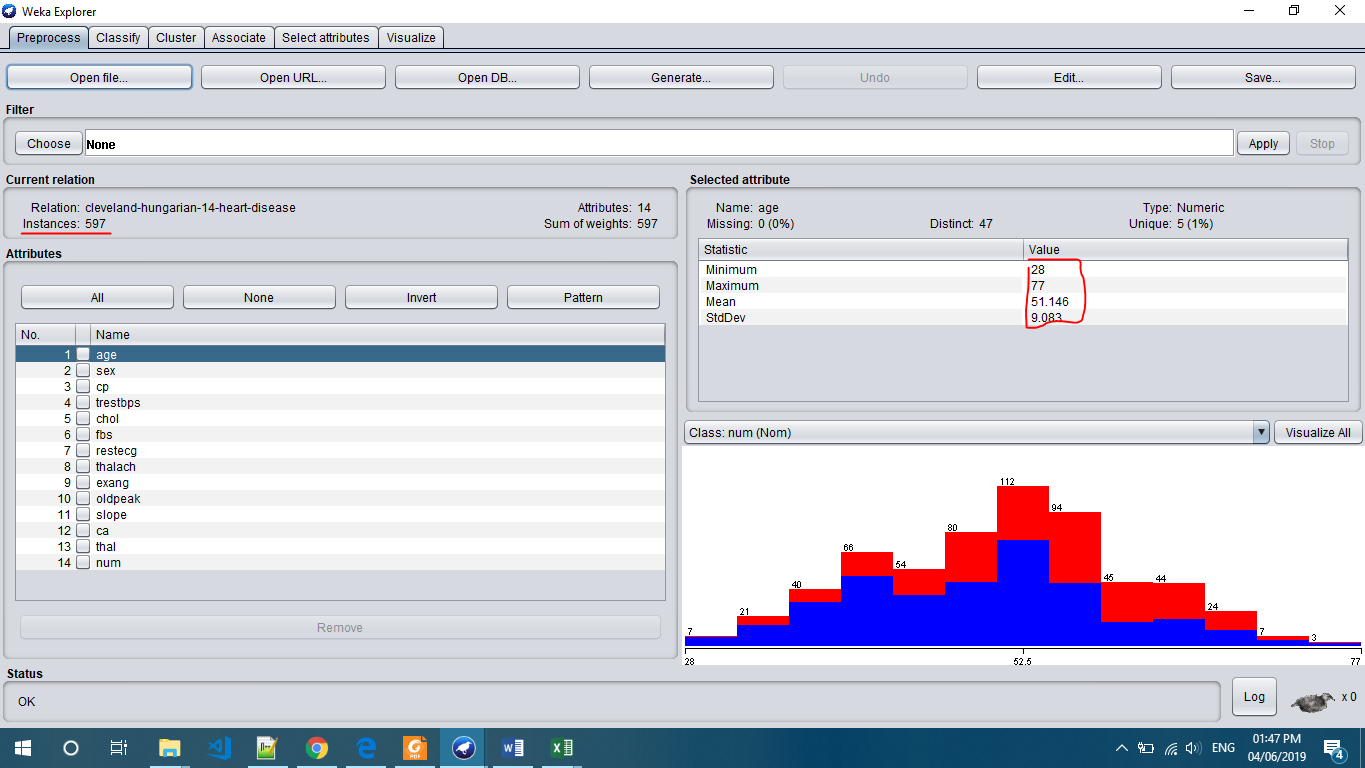
1. randomSeed: đặt mầm (seed) ngẫu nhiên để lấy mẫu
2. biasToUniformClass (thông số này chỉ có khi chọn mục supervised): giá trị 0 – phân phối của lớp như hiện trạng (như bộ dữ liệu đầu). Giá trị 1 – phân phối lớp của mẫu là phân phối chuẩn (uniform).
3. noReplacement: True – with replacement (lấy tuple dữ liệu ra và trả lại nó trong bộ dữ liệu D, nghĩa là có thể nó sẽ vẫn có thể được rút ra trong lần lấy tuple sau). False – without replacement (lấy tuple dữ liệu ra (với xác suất rút 1/N) và không trả nó lại trong bộ dữ liệu D, nghĩa là mỗi tuple chỉ xuất hiện 1 lần trong mẫu).
4. sampleSizePercent: kích thước của mẫu so với bộ dữ liệu gốc bao nhiêu phần trăm.
5. invertSelection: lấy mẫu với các bộ ngược lại (inverse) so với mẫu lấy được (chỉ áp dụng với mẫu không có lặp – without replacement).

 (6.2)

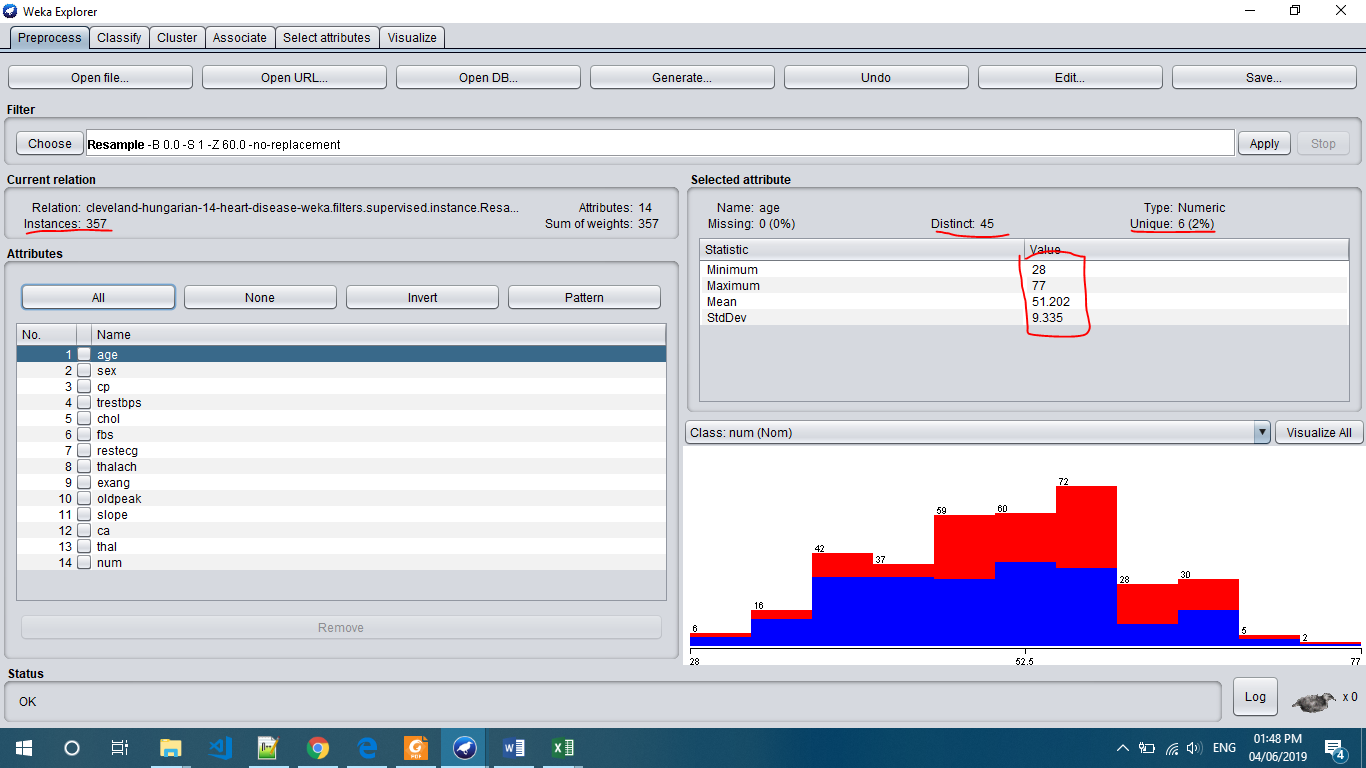
* B3: Sau khi chọn các thông số theo nhu cầu, nhấn nút Apply để lấy mẫu (sampling).

Ví dụ: ở đây nhóm lấy mẫu vói các thông số: noReplacement = True, sampleSize = 60.

 (6.3)



(Ảnh bộ dữ liệu gốc (6.4))



(Ảnh: mẫu lấy được với các thông số như trên)

1. Trong Weka, ta có thể thực hiện 2 phương pháp chính: Simple Random Sample Without Replace (SRSWOR) và Simple Random Sample With Replacement (SRSWR) bằng cách đặt thông số option noReplacement là True (không lặp) hoặc False (có lặp).