**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH**

**🙥🟊🙧**

**BÀI TẬP 1: TIỀN XỬ LÝ DỮ**

**LIỆU VỚI WEKA**

1. **THÔNG TIN SINH VIÊN**

Họ và tên: **TRẦN NHẬT HUY**

Mssv: **1612272**

Email: [**nhathuy13598@gmail.com**](mailto:nhathuy13598@gmail.com)

Sđt: **0354 878 677**

1. **BẢNG BÁO CÁO CÔNG VIỆC**

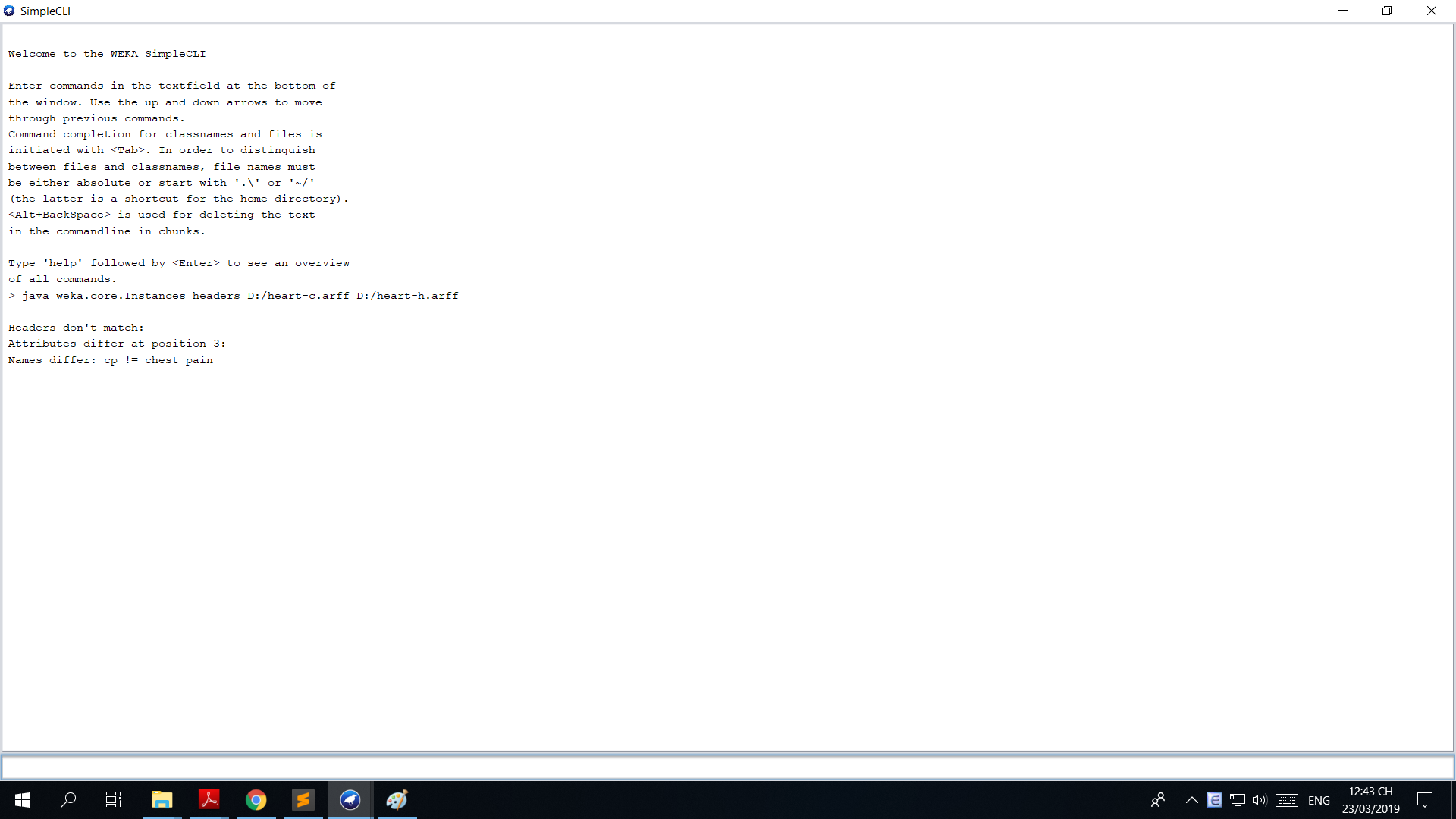
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **CÁC CÂU HỎI** | **MỨC ĐỘ HOÀN THÀNH** | **GHI CHÚ** |
| 1 | a. Định nghĩa sự hợp nhất dữ liệu | 100% |  |
| b. Có vấn đề nhận hiện thực thể hay không? Giải quyết (nếu có) | 100% |  |
| c. Có vấn đề dữ liệu dư thừa không? Giải quyết (nếu có) | 100% |  |
| d. Có mâu thuẫn dữ liệu không? Giải quyết (nếu có) | 100% |  |
| e. Tích hợp 2 dataset | 100% |  |
| f. Chụp lại màn hình | 100% |  |
| 2 | a. Xem thuộc tính age và trả lời các câu hỏi | 100% |  |
| b. Liệt kê five-number summary của thuộc tính age | 100% |  |
| c. Bao nhiêu thuộc tính số, có thứ tự, rời rạc/danh sách? | 100% |  |
| d. Giải thích đồ thị trong Explorer | 100% |  |
| e. Dán ảnh chụp các đồ thị vào bài làm | 100% |  |
| f. Nhận xét về đồ thị | 100% |  |
| g. Dán đồ thị bạn cho rằng có khả năng đoán bệnh tim tốt nhất | 100% |  |
| h. Những cặp thuộc tính nào tương quan? | 100% |  |
| 3 | a. Có bao nhiêu thuộc tính trong dataset? | 100% |  |
| b. Liệt kê các phương pháp lọc thuộc tính | 100% |  |
| c. So sánh các phương pháp trong textbook và weka | 100% |  |
| 4 | a. Dữ liệu thiếu | 80% | Chưa cài đặt một phương pháp trong weka |
| b. Dữ liệu nhiễu | 100% |  |
| c. Dữ liệu tạp | 100% |  |
| d. Lưu dataset đã làm sạch | 100% |  |
| 5 | a. Xây dựng thuộc tính | 100% |  |
| b. Chuẩn hóa | 100% |  |
| c. Chọn 1 phương pháp chuẩn hóa | 100% |  |
| d. Lưu dataset đã chuẩn hóa | 100% |  |
| 6 | Lấy mẫu | 100% |  |

1. **CHI TIẾT BÀI LÀM**
2. **CHUẨN BỊ DỮ LIỆU – TÍCH HỢP DỮ LIỆU (INTEGRATION)**
3. *Định nghĩa sự tích hợp dữ liệu*

Sự tích hợp dữ liệu **(Data Integration)** là kết hợp các dữ liệu từ nhiều nguồn khác nhau để tạo thành một kho dữ liệu mạch lạc, các nguồn này có thể bao gồm nhiều **cơ sở dữ liệu** **(database)**, **khối dữ liệu (data cube)** hoặt là **file phẳng (flat file)**.

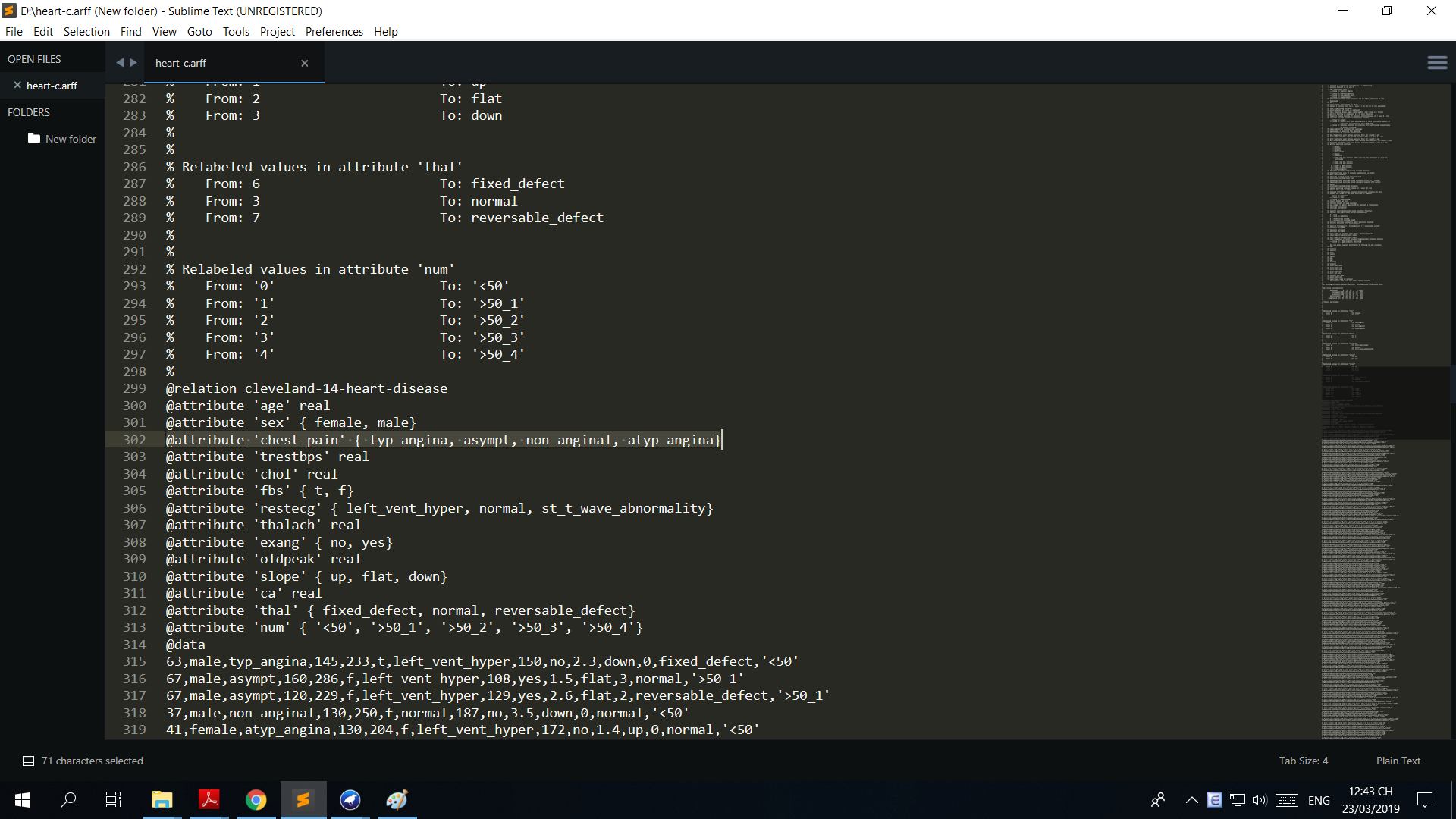
1. *Có vấn đề về nhận diện thực thể (entity identification) trong 2 dataset này hay không? Nếu có, giải quyết như thế nào?*

Trong 2 data này có vấn đề về nhận diện thực thể. Ta sẽ kiểm tra bằng lệnh trong WEKA và đây là kết quả:



Hình 1: Bị lỗi attributes cp != chest\_pain

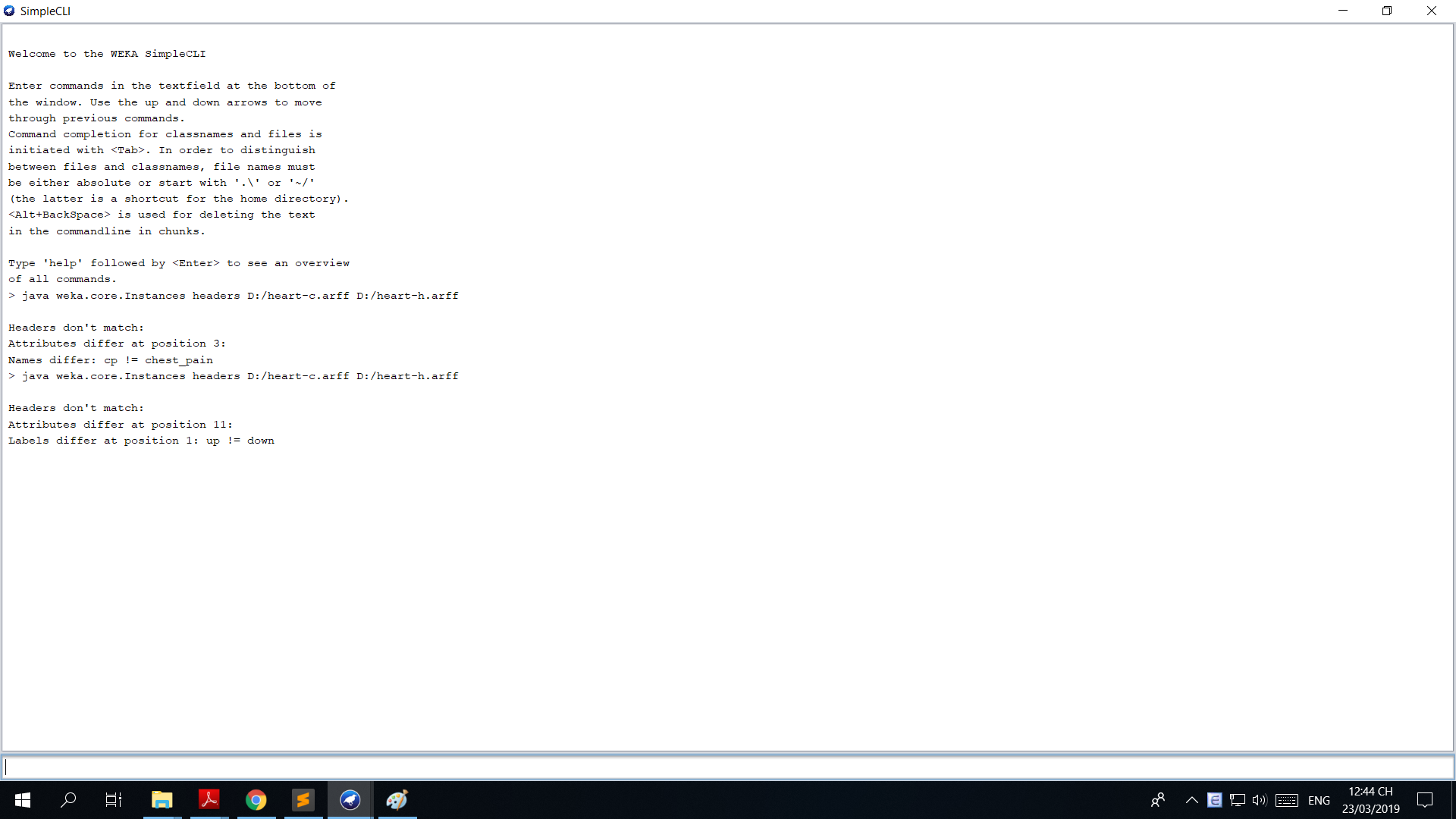
Ta sẽ tiến hành sửa lại thuộc tính **cp** thành **chest\_pain**



Hình 2: Sửa lại trong file heart-c.arff

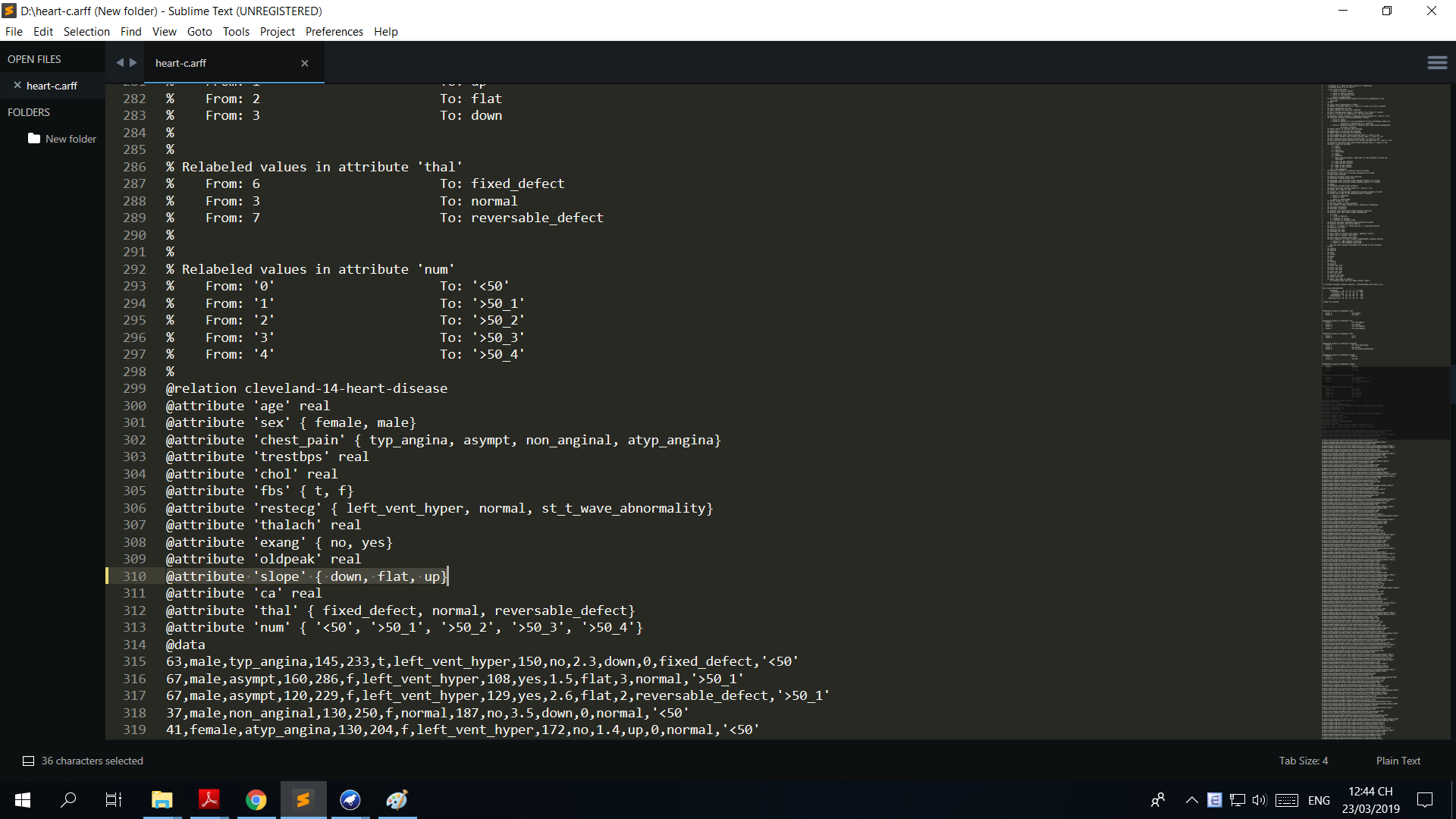
Sửa lại thuộc tính **cp** trong file **heart-c.arff** thành **chest\_pain**

Kiểm tra lại xem còn bị lỗi hay không



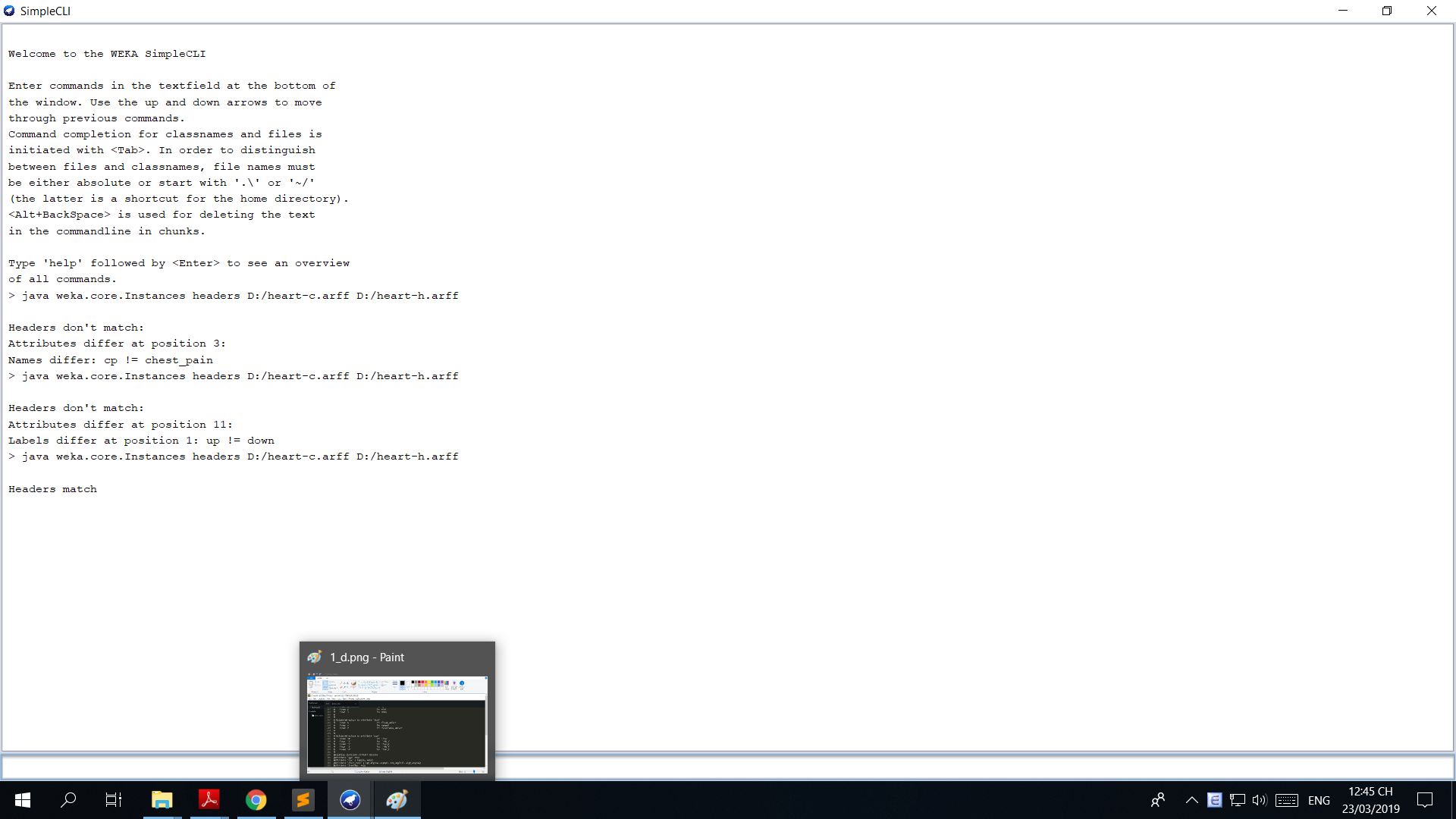
Hình 3: Lỗi Labels differ at position 1 up != down

Tiến hành sửa lỗi, lần này chúng ta chỉ cần thay đổi thứ tự của **up** và **down** cho phù hợp



Hình 4: Sửa lại thứ tự trong file heart-c.arff

Tiến hành kiểm tra lại



Hình 5: Lần này không có lỗi nên có thông báo Header match

1. *Có vấn đề dữ liệu dư thừa (redundancy) trong 2 dataset này hay không? Nếu*

*có, giải quyết như thế nào?*

Trong 2 data này có sự **dư thừa dữ liệu (redundancy)**, cụ thể là sự **trùng lặp dữ liệu (duplication)**. Chúng ta sẽ giải quyết vấn đề này bằng cách sử dụng bộ lọc mà Weka cung cấp sẵn sau khi đã hợp nhất 2 dataset lại với nhau

1. *Có sự mâu thuẫn dữ liệu (data value conflicts) trong 2 dataset này hay không? Nếu có, giải quyết như thế nào?*

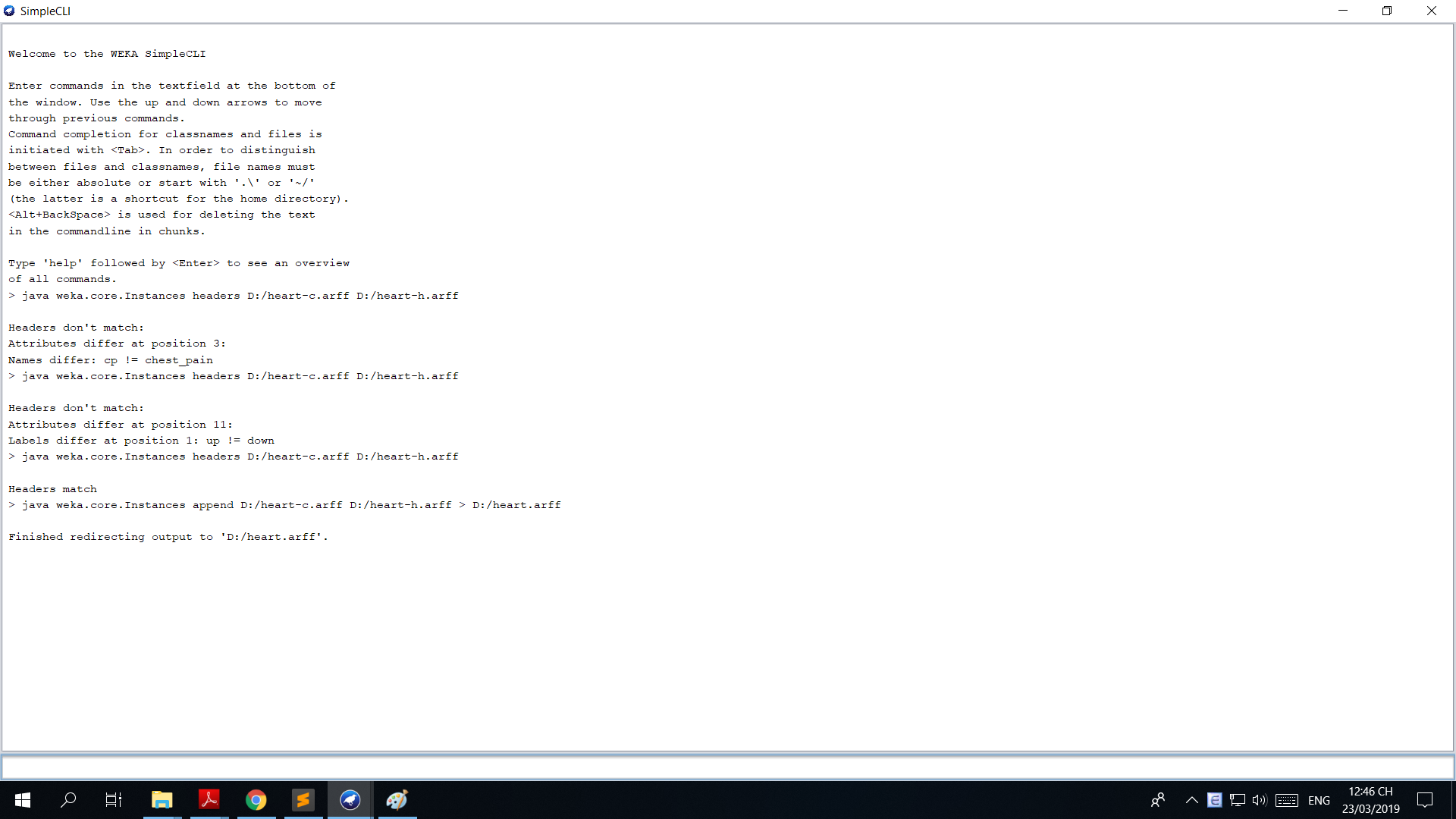
Trong 2 dataset này có **mâu thuẫn dữ liệu (data value conflicts)**, cụ thế là giá trị của attribute **slope** mà chúng ta đã chỉnh sửa ở trên

1. *Tích hợp 2 dataset này lại thành 1 dataset để chuẩn bị cho các câu hỏi tiếp*

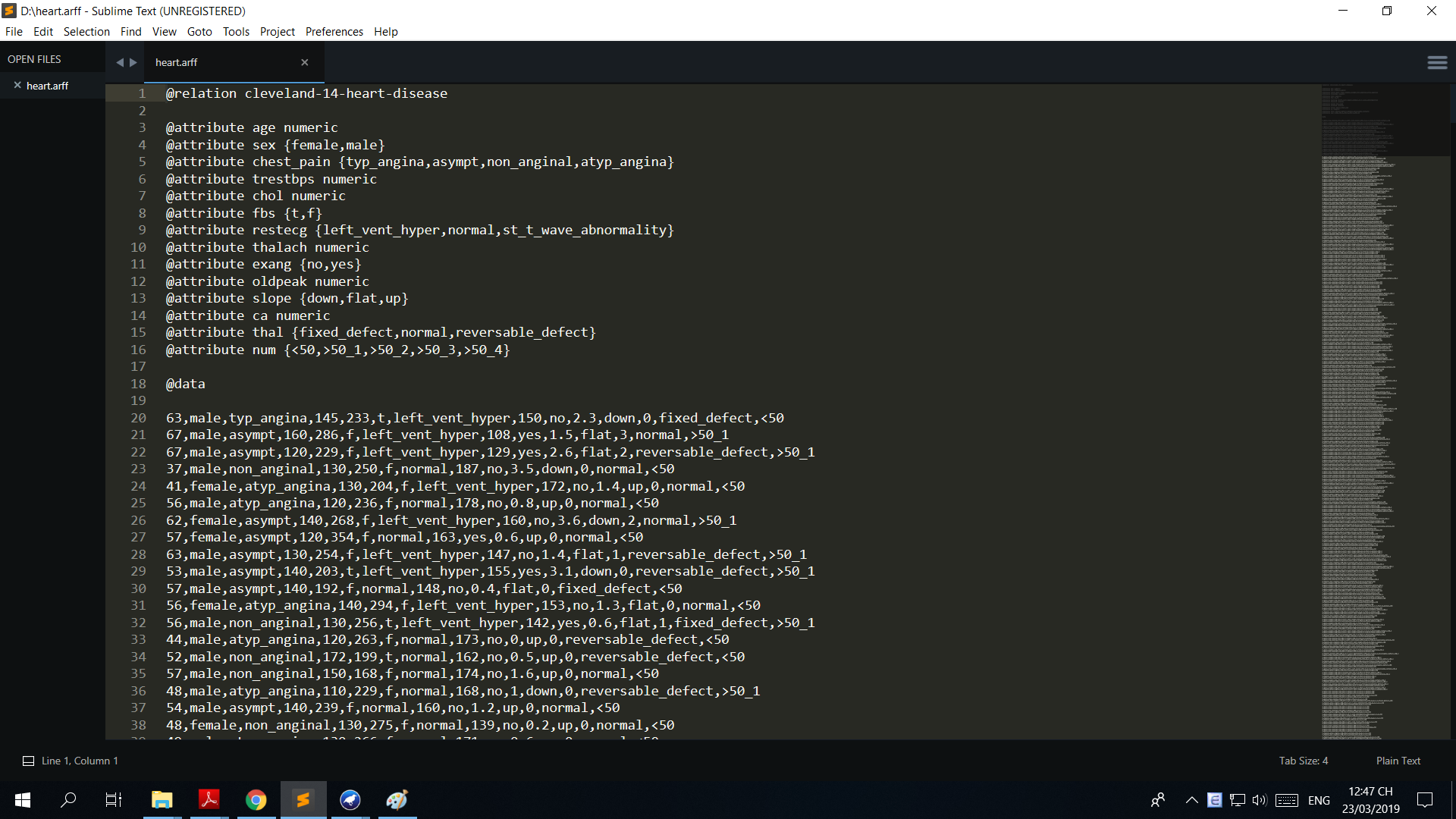
*theo. Nạp dataset sau khi tích hợp vào Explorer. Bạn có bao nhiêu mẫu? Bao*

*nhiêu thuộc tính?*

Để tích hợp 2 dữ liệu lại với nhau ta sẽ sử dụng lệnh **java weka.core.Instances append** trong **Simple CLI**



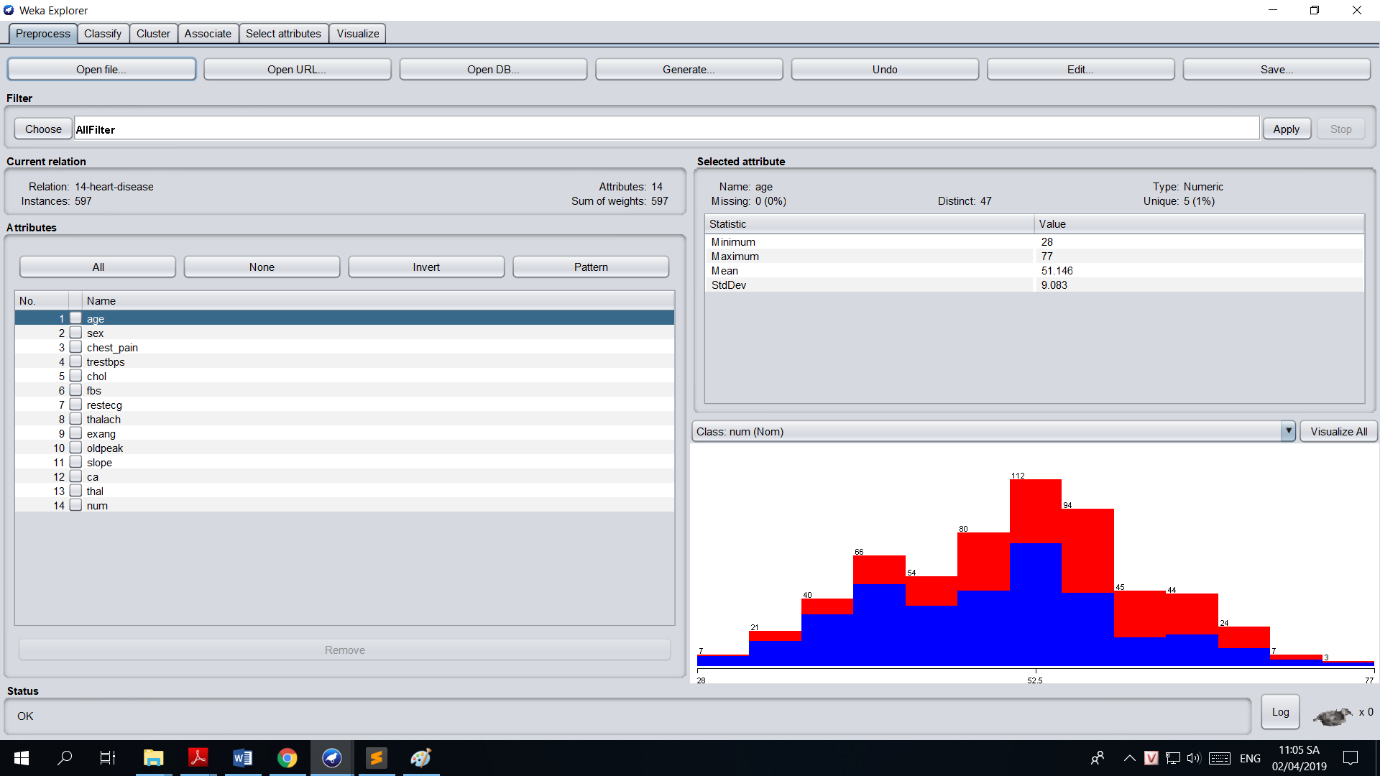
Hình 6: Tạo file heart.arff bằng lệnh append



Hình 7: File heart.arff sau khi sử dụng lệnh append

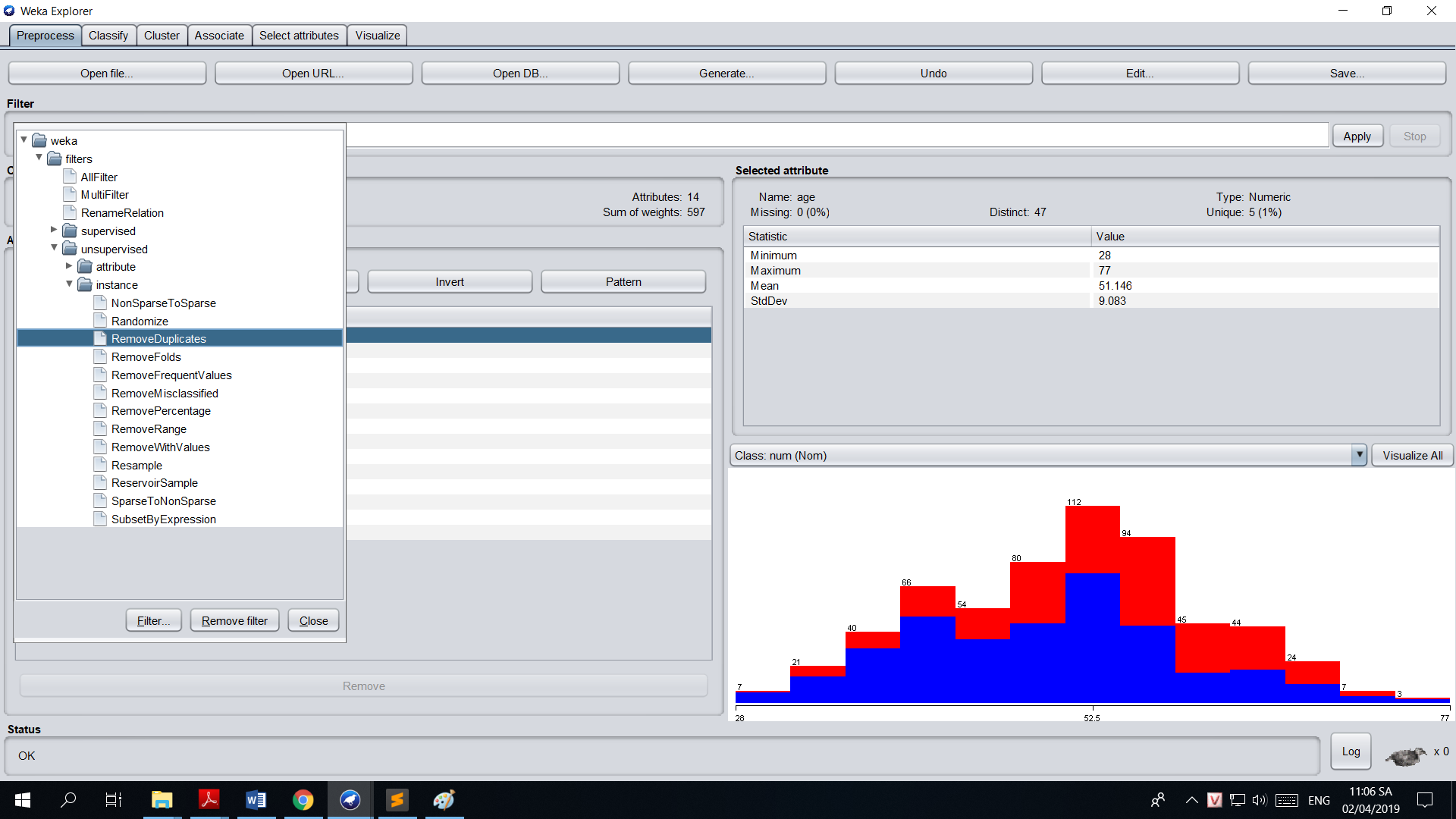
Ta sửa lại tên quan hệ là **14-heart-disease**

Load dataset vào Explorer của Weka



Hình 8: Dataset trong Weka

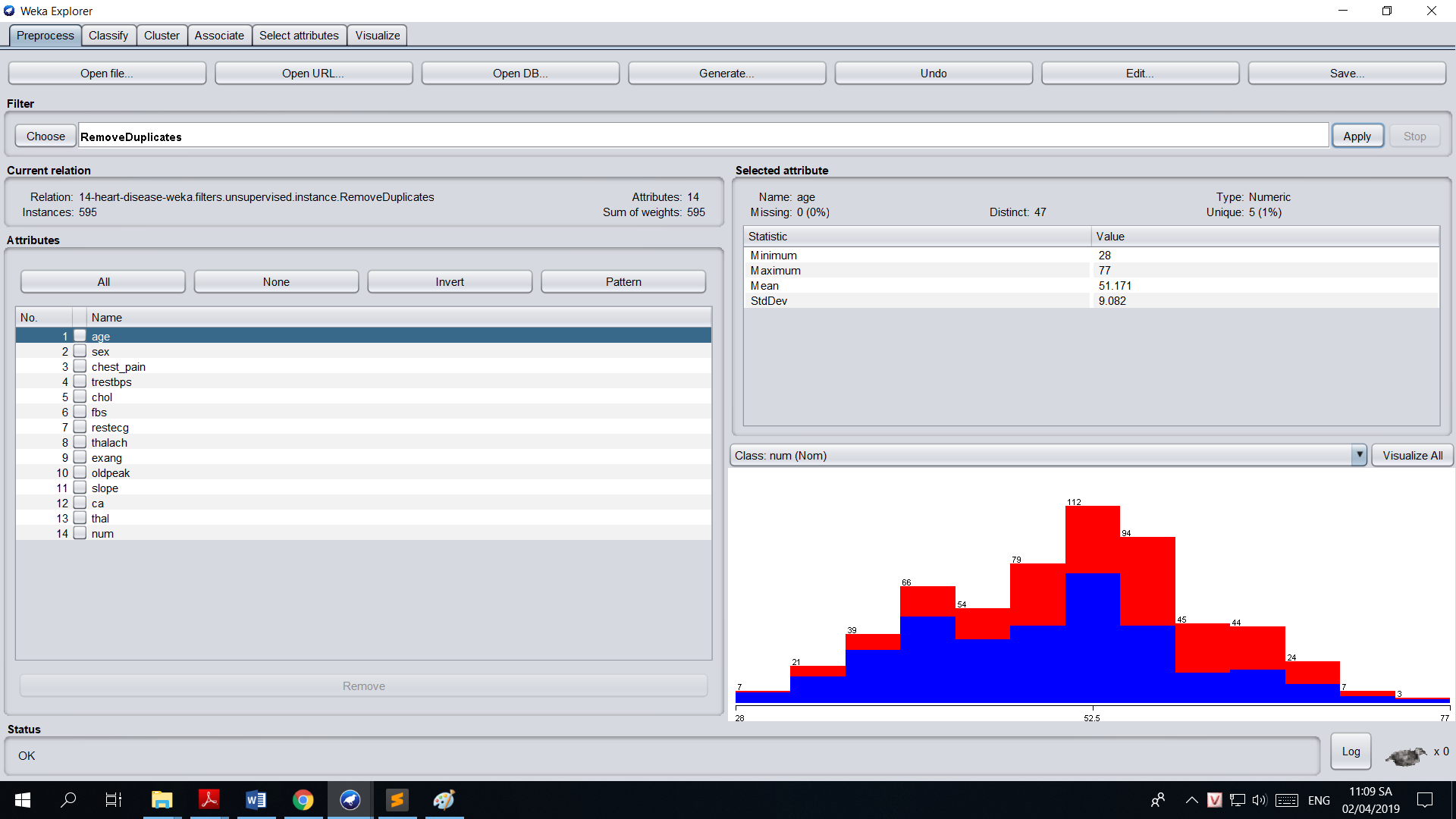
Ta tiến hành loại bỏ **Duplicate Instances** bằng bộ lọc như trong hình



Hình 9: Dùng bộ lọc RemoveDuplicate

1. *Chụp lại màn hình của cửa sổ Explorer của bạn.*

Ta chọn **Apply** và đây là kết quả

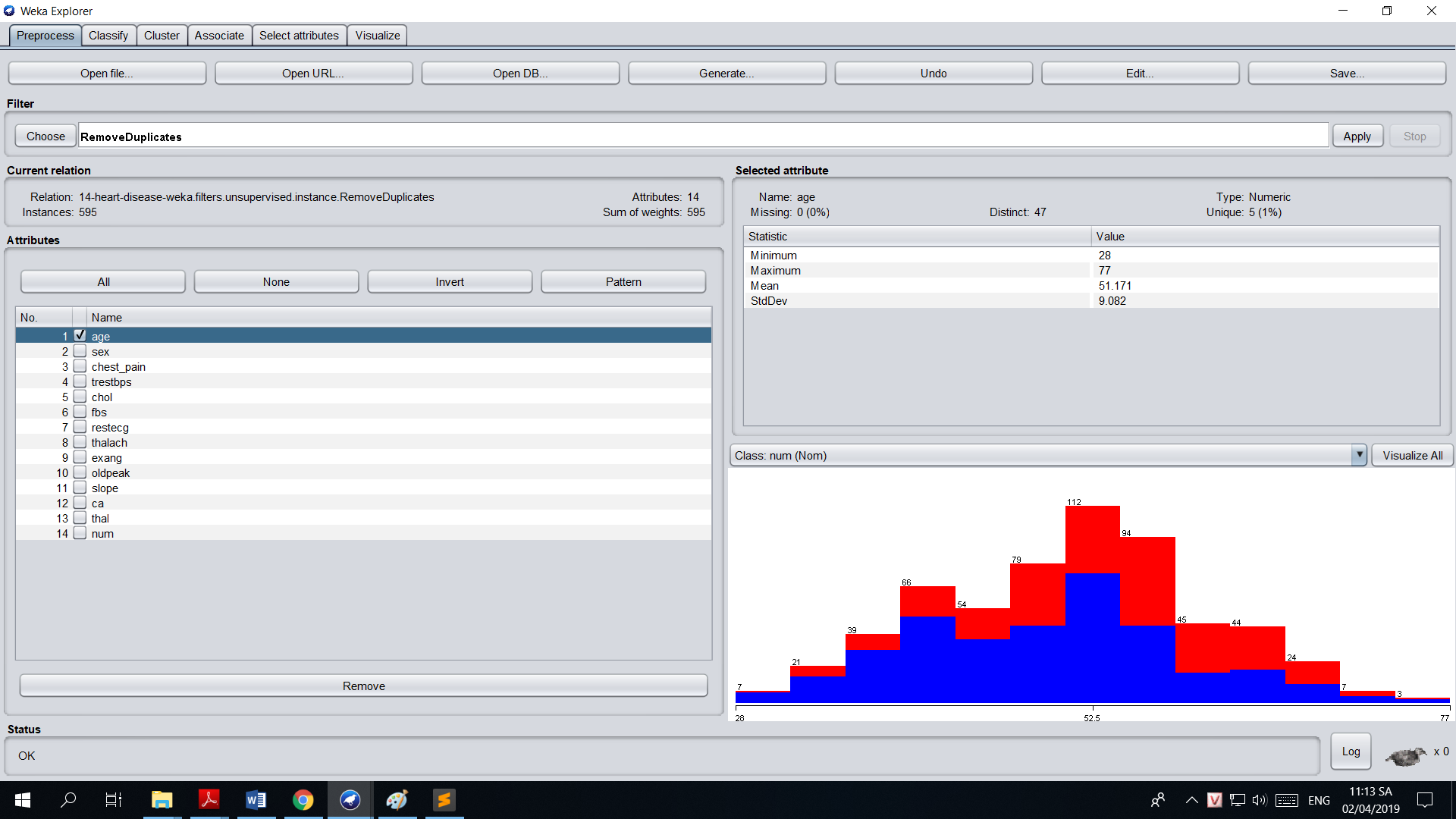


Ban đầu có **597 instances**, sau khi lọc chúng ta còn **595**

Nhấn **save** để lưu lại dataset này cho các bước sau

1. **TÓM TẮT MÔ TẢ DỮ LIỆU – DESCRIPTIVE DATA SUMMARIZATION**
2. *Trong* ***tab Preprocess****, xem xét thuộc tính age và trả lời câu hỏi: trung bình,*

*độ lệch chuẩn, giá trị nhỏ nhất, lớn nhất của nó là gì?*



Hình 10: Thuộc tính Age

**Trung bình (Mean)** là: 51.171

**Độ lệch chuẩn (StdDev)** là: 9.082

**Giá trị nhỏ nhất (Minimum)** là: 28

**Giá trị lớn nhất (Maximum)** là: 77

1. *Liệt kê five-number summary của thuộc tính này. Weka có cung cấp những con số này hay không?*

Min: 28

Q1: 44

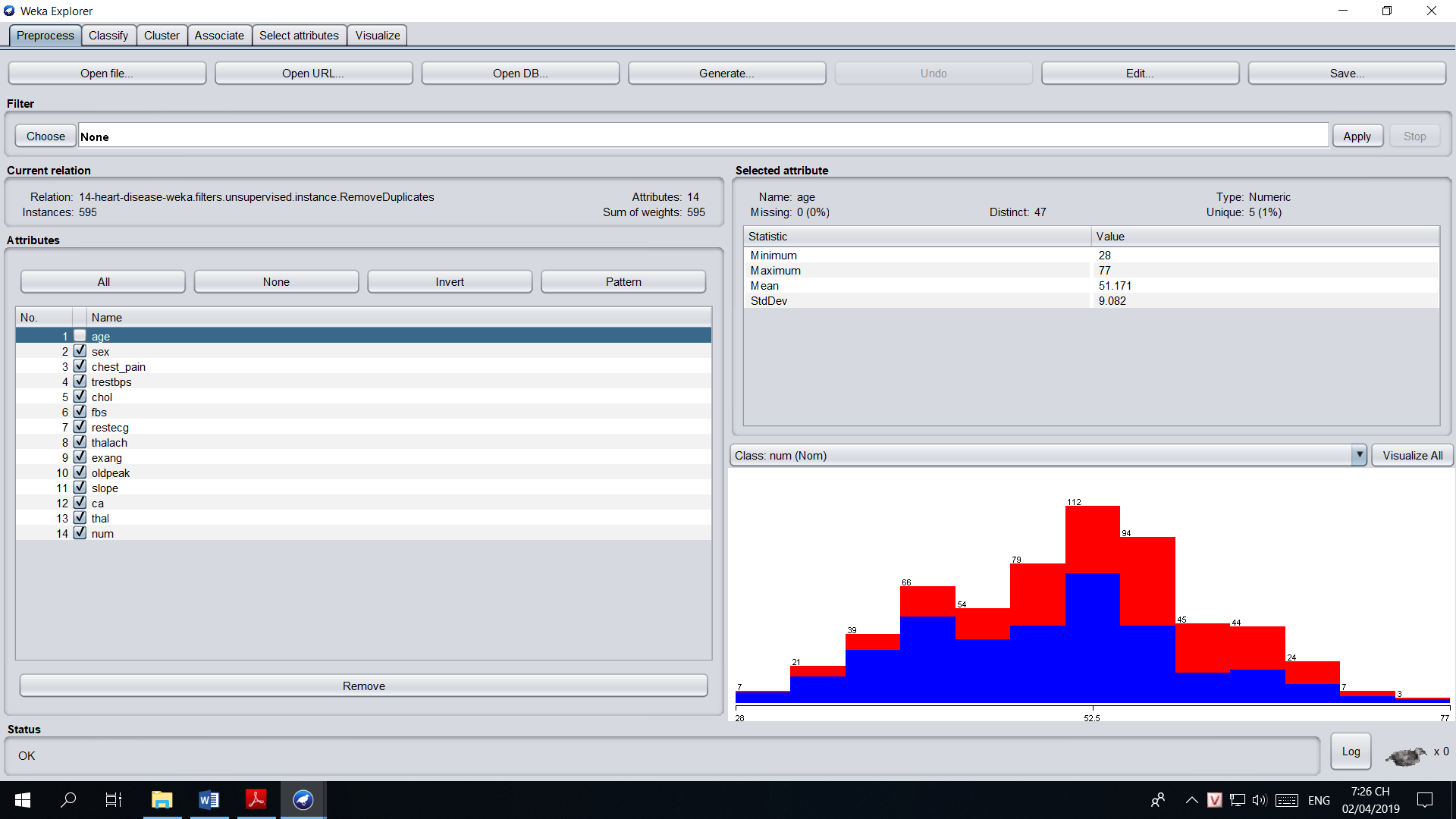
Median: 52

Q2: 88

Max: 77

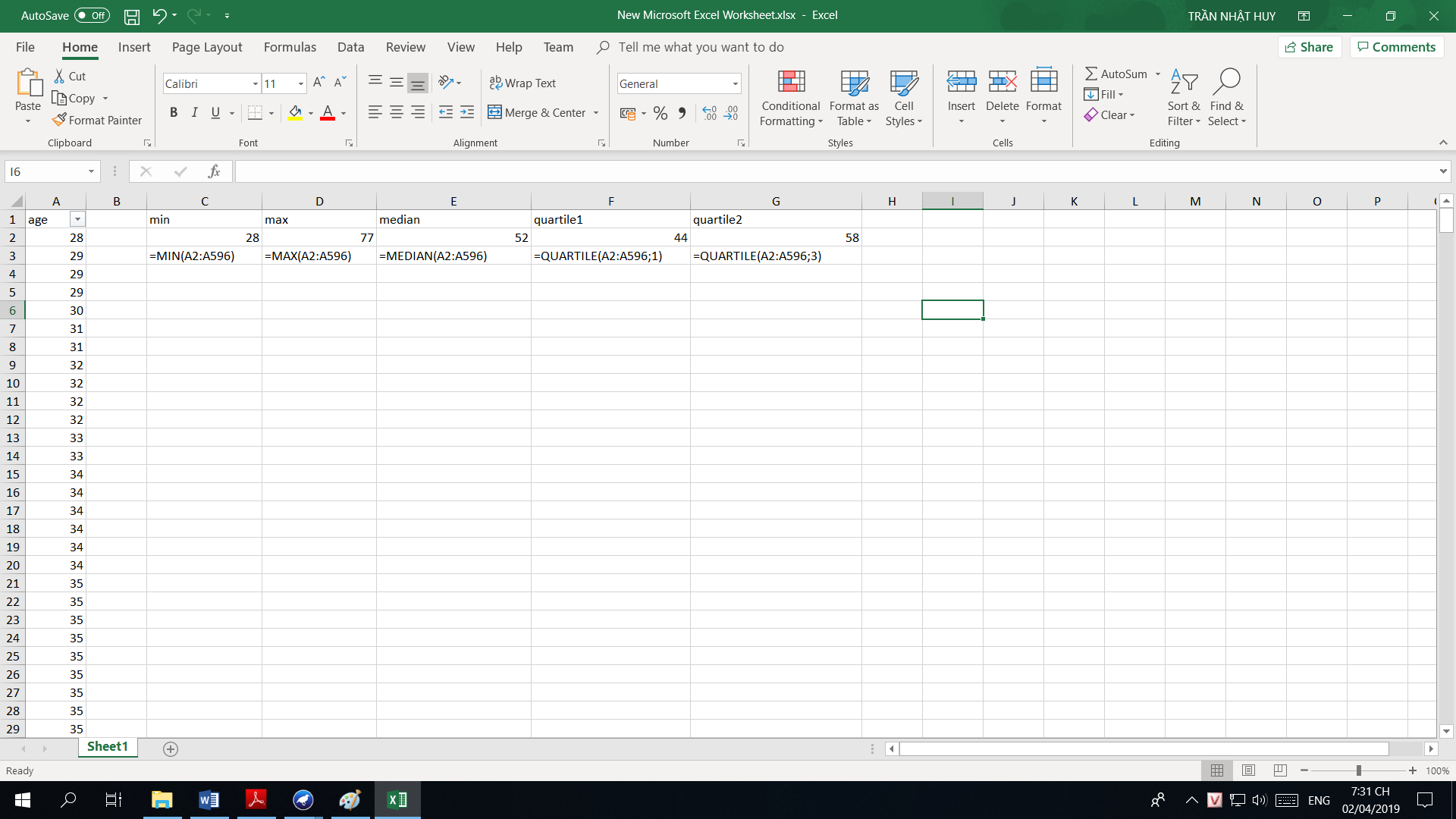
Weka chỉ cung cấp 1 phần các con số này cụ thể là **min**, **max** còn **Q1**, **Median**, **Q2** thì chúng ta phải tự tìm. Ở đây, em sử dụng **Excel** để tìm.

Ta chọn thuộc tính **age** để giữ lại, xóa attribute khác và lưu thành một file **heart\_age.arff**



Hình 11: Nhấn Remove để loại bỏ và Save lại

Ta sao chép data trong file **heart\_age.arff** vào excel để thao tác như trong hình



Hình 12: Thao tác làm trong Excel

1. *Cho biết thuộc tính nào là* ***số (numeric)****, thuộc tính nào là có* ***thứ tự (ordernal)*** *và thuộc tính nào là* ***rời rạc/danh sách (categorical/nomial).***

Thuộc tính số (numeric): **age, trestbps, chol, thalach, oldpeak, ca.**

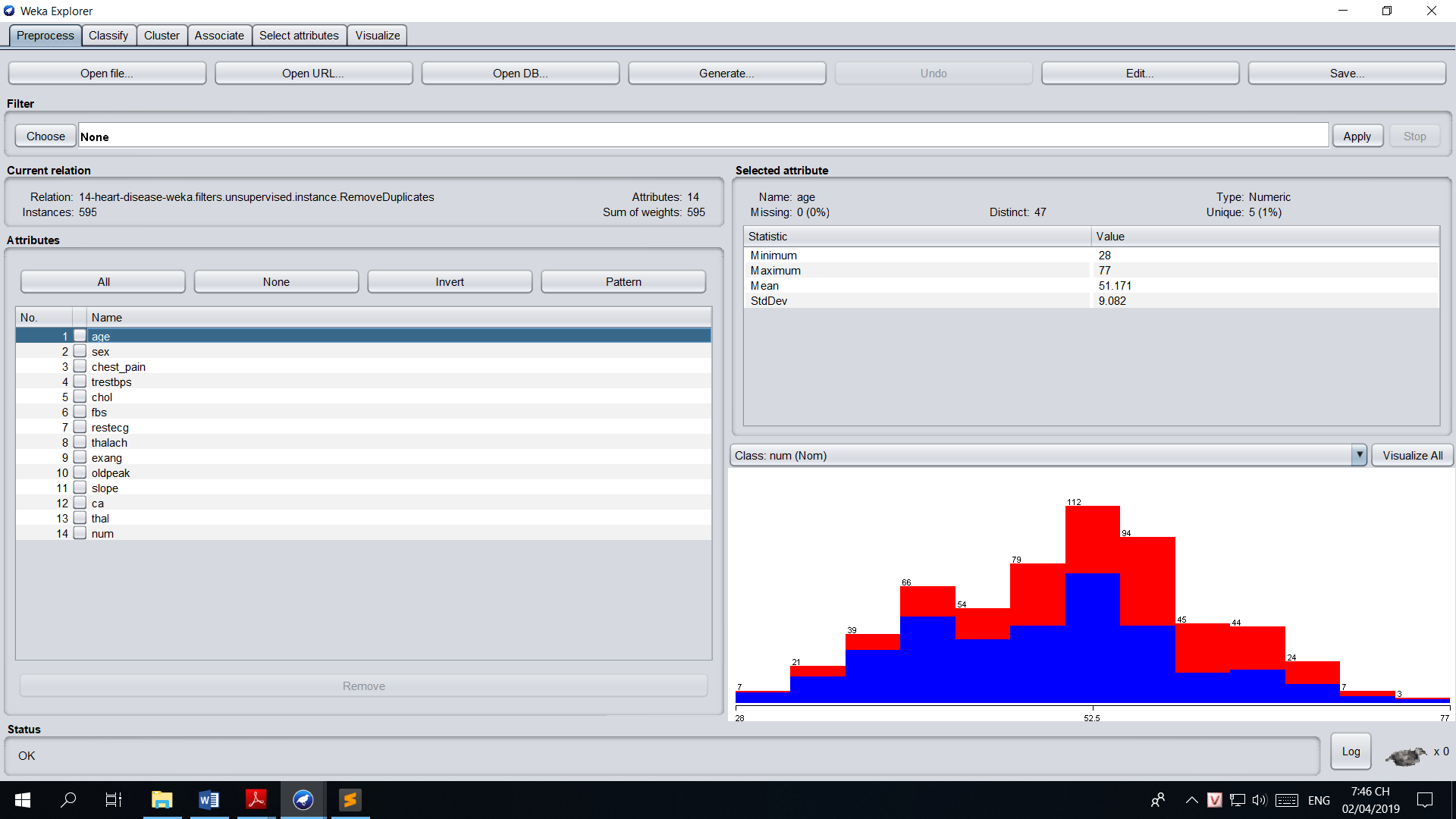
Thuộc tính có thứ tự (ordernal): Không có

Thuộc tính rời rạc/danh sách (catergorical/nominal): **sex, chest\_pain, fbs, restecg, exang, slope, thal, num**

1. *Giải thích ý nghĩa của đồ thị trong cửa sổ Explorer. Bạn đặt tên cho đồ thị này là gì? Màu xanh và màu đỏ có nghĩa gì (chú ý các pop-up hiện lên khi di*

*chuyển chuột trên đồ thị). Đồ thị này biểu diễn cho cái gì?*

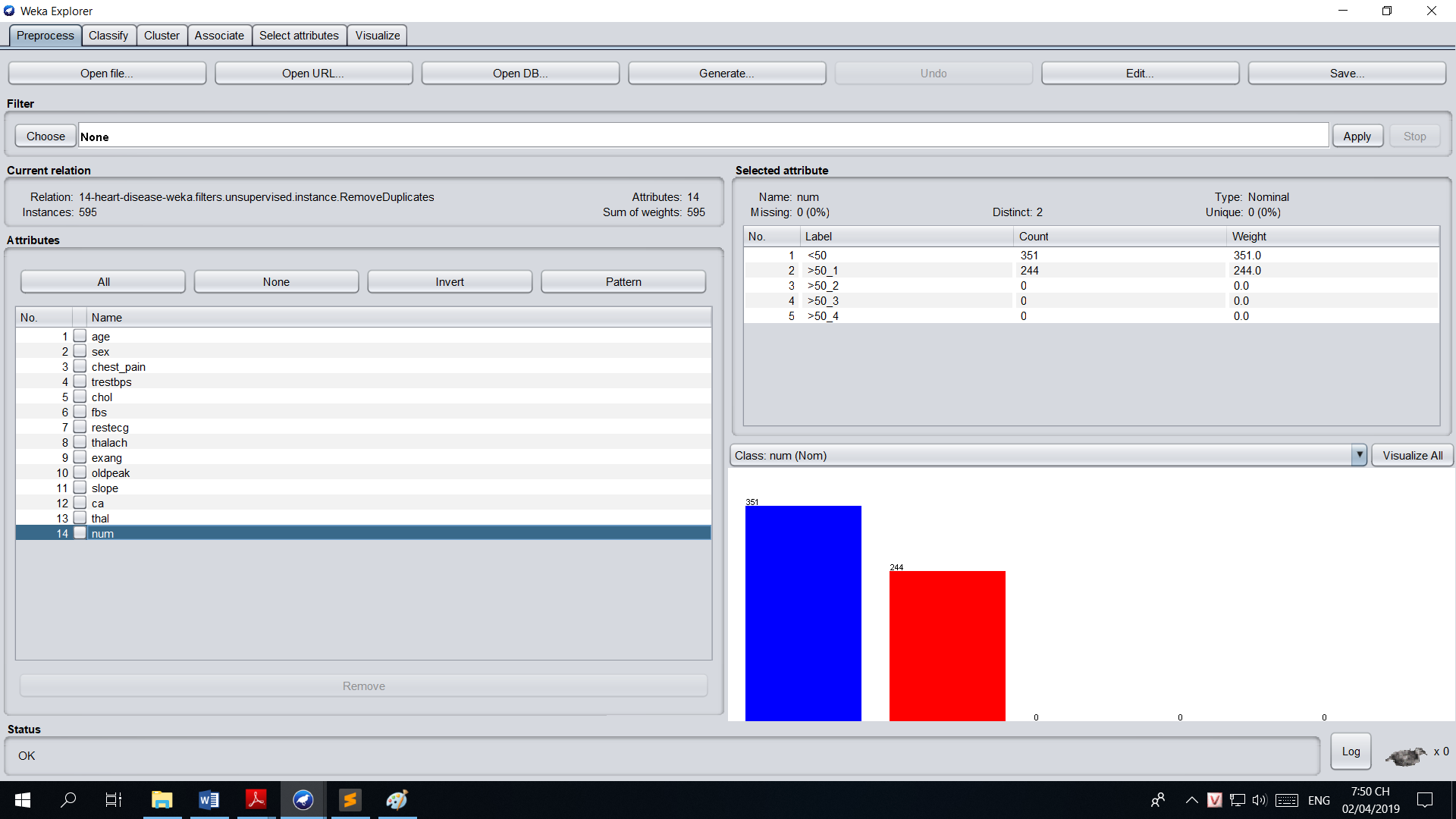
Đồ thị trong cửa sổ **Explorer** thể hiện phân bố của kết quả cần dự đoán theo một attribute cụ thể nào đó. Em đặt tên là **histogram**



Hình 13: Phân bố của kết quả dựa theo thuộc tính age

Màu xanh thể hiện **class num** có value là **< 50**

Màu đỏ thể hiện class **num** có value là **> 50\_1**

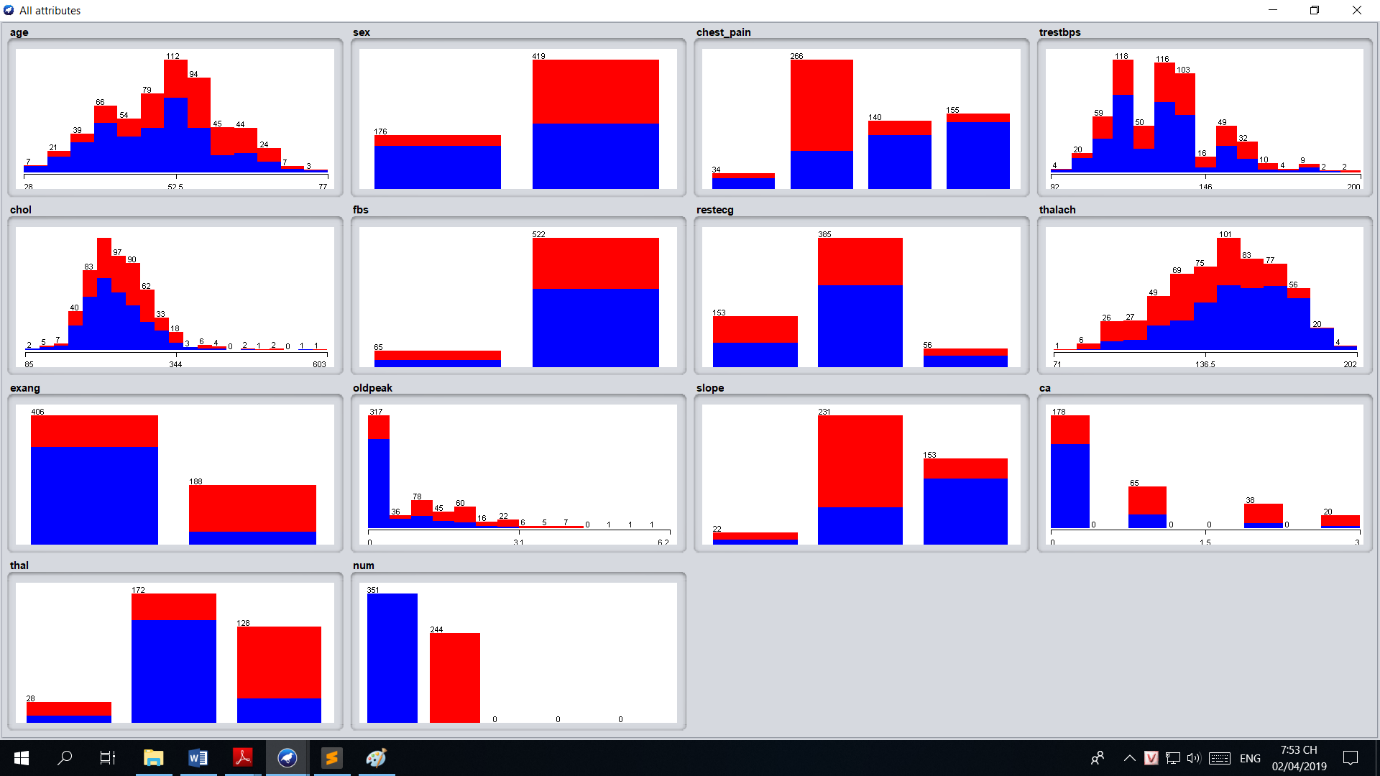


Hình 14: Chọn thuộc tính num để xác định ý nghĩa

1. *Lần lượt xem xét các thuộc tính khác của dataset dưới dạng đồ thị. Dán các*

*ảnh chụp màn hình vào bài làm.*

Ta sẽ nhấn nút **Visualize all** để hiển thị tất cả các đồ thị



Hình 15: Đồ thị cho tất cả các attribute

1. *Nhận xét của bạn từ những đồ thị đó?*

Thuộc tính **num** sẽ có phân bố gần giống với các attribute được chọn

1. *Chuyển sang* ***tab Visualize****. Thuật ngữ sử dụng trong textbook để đặt tên cho*

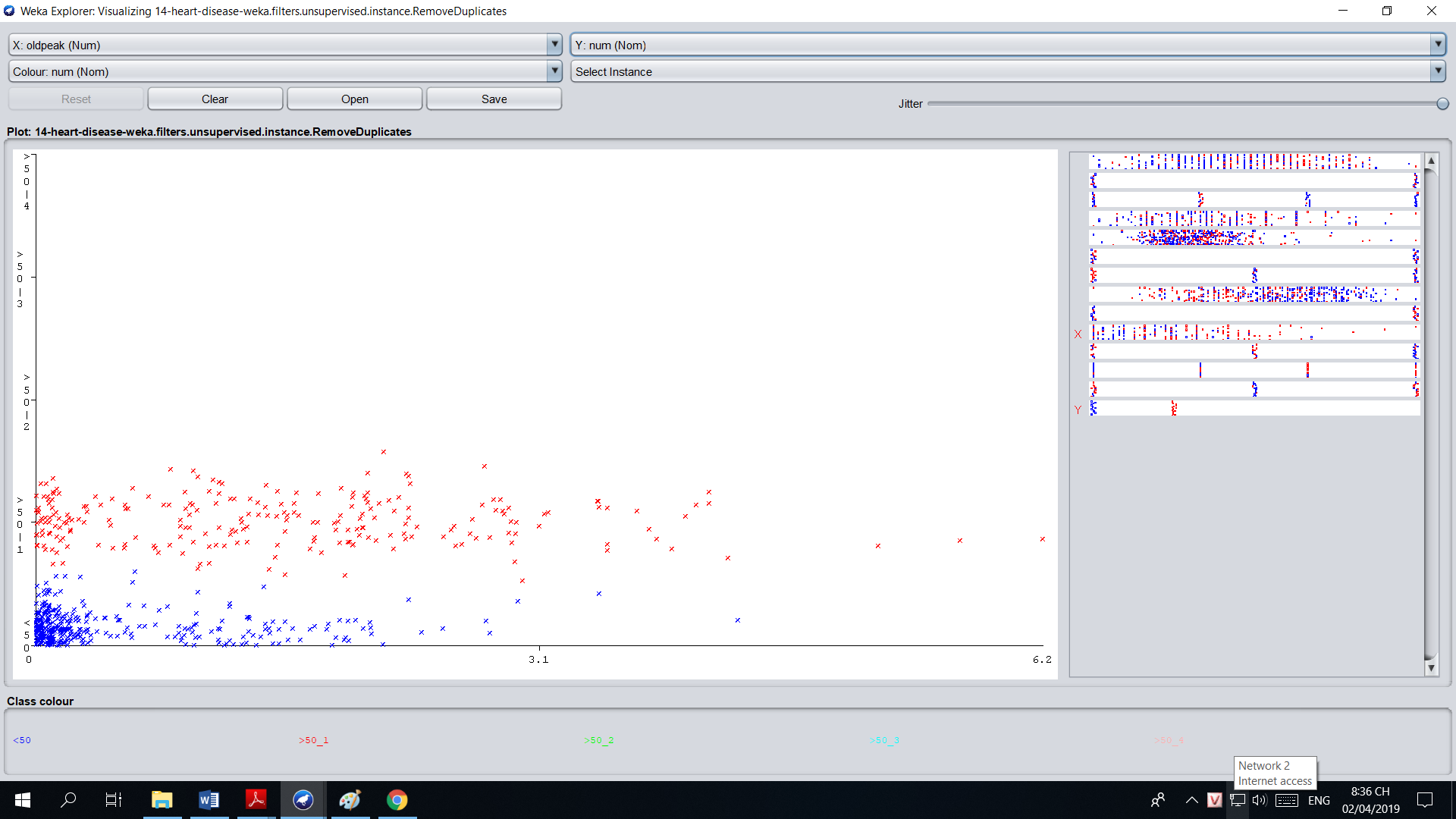
*các đồ thị là gì? Chọn* ***jitter*** *tối đa, chú ý cột num (cột cuối cùng), theo bạn*

*các thuộc tính nào có vẻ như dẫn đến bệnh tim nhiều nhất? Dán vào bài làm*

*hình ảnh đồ thị của thuộc tính mà bạn cho rằng có khả năng dự đoán bệnh tim tốt nhất* ***(Y)*** *như là một hàm của* ***num(X)****.*

Thuật ngữ sử dụng trong textbook là **Scatter plot matrix**

Theo em thì thuộc tính **oldpeak** có vẻ dẫn đến bệnh tim nhiều nhất



Hình 16: Đồ thị của num(X) và Oldpeak(Y)

1. *Có những cặp thuộc tính khác nhau nào có vẻ như tương quan với nhau*

*không?*

Có các thuộc tính tuy khác nhau nhưng có vẻ tương quan với nhau như:

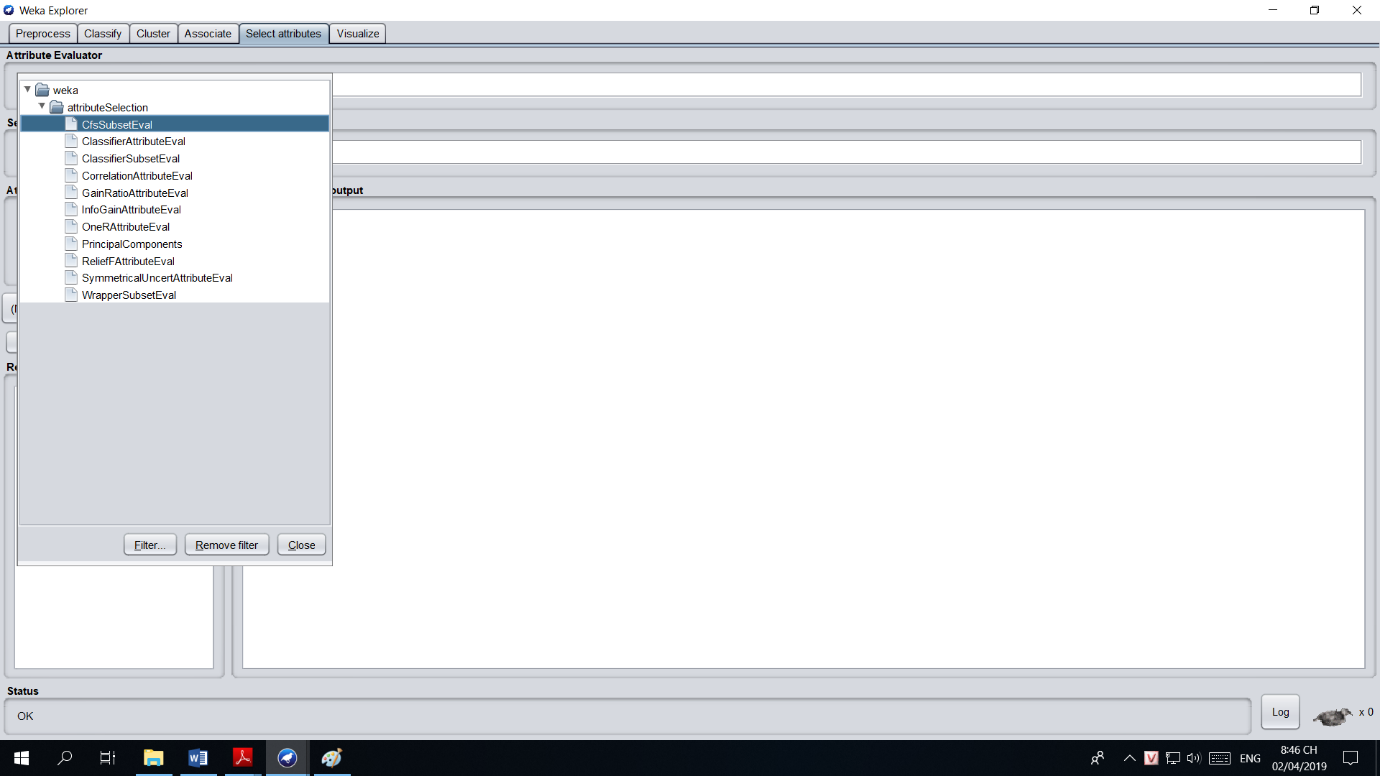
* thal, cal, slope, restecg
* oldpeak, thalach, chol, tresbps, chest\_pain, age
* sex, exang

1. **CHUẨN BỊ DỮ LIỆU – CHỌN LỌC DỮ LIỆU**
2. *Bạn hãy cho biết có bao nhiêu thuộc tính trong những dataset trước khi xử lý?*

Có tất cả 14 thuộc tính trong dataset với thuộc tính **num** là cần dự đoán

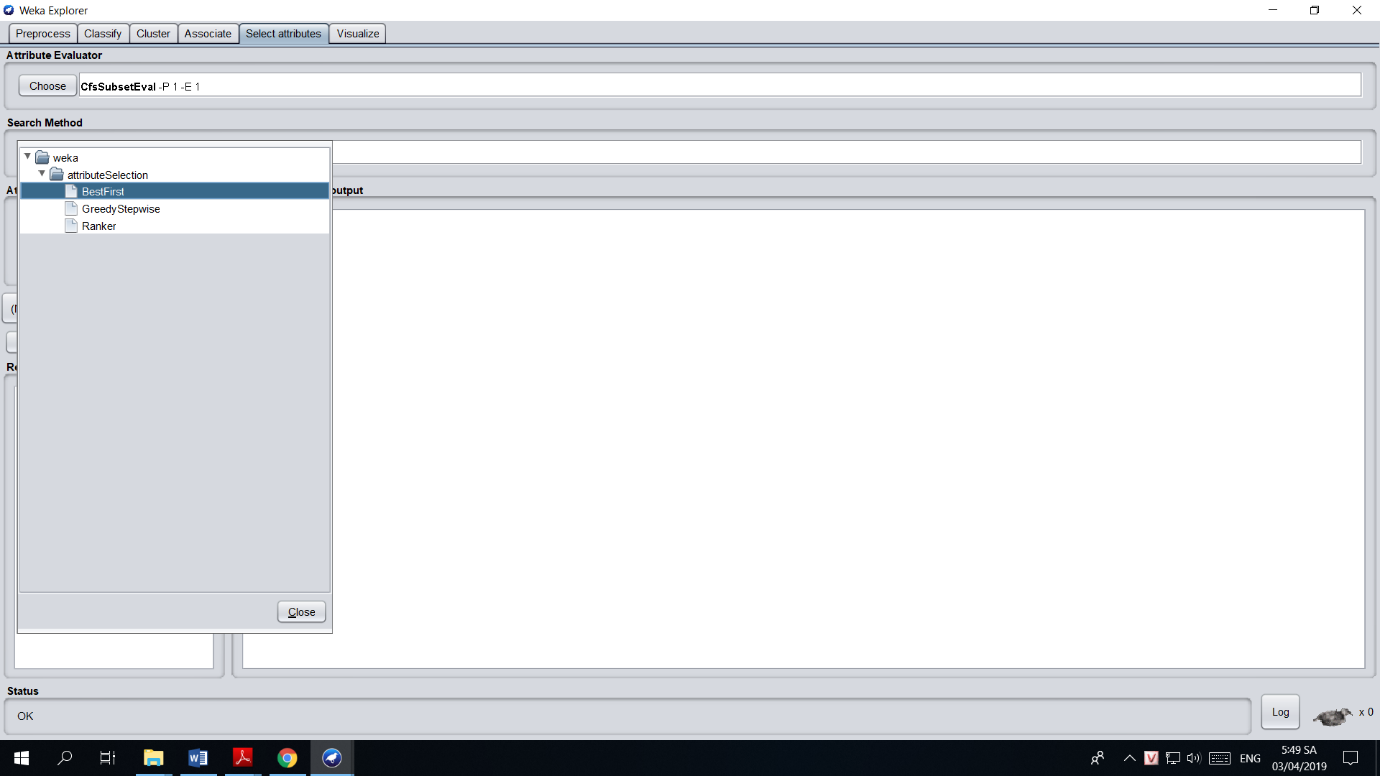
1. *Sử dụng* ***tab Select attributes****. Liệt kê những lựa chọn khác nhau của Weka*

*để chọn lọc thuộc tính, giải thích ngắn gọn từng phương pháp.*



Hình 17: Các phương pháp Attribute Evaluator

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN PHƯƠNG PHÁP** | **MỤC ĐÍCH** |
| 1 | **CfsSubsetEval** | Đánh giá khả năng dự đoán của từng attribute riêng lẻ và mức **độ dư thừa (redundancy)** giữa chúng, ưu tiên tập các attribute mà có quan hệ cao với **lớp (class)** nhưng có sự tương quan lẫn nhau thấp. Các **giá trị thiếu (Missing values)** sẽ được xem như là giá trị rời rạc hoặc giá trị của nó sẽ phụ thuộc vào phân bố của các giá trị |
| 2 | **GainRatioAttributeEval** | Đánh giá những thuộc tính bằng cách đo **tỉ lệ tăng (gain ratio)** của thuộc tính đó cho **class**. |
| 3 | **InfoGainAttributeEval** | Đánh giá những thuộc tính bằng cách đo **độ tăng thông tin (Information Gain)** cho class. Phương pháp này sẽ rời rạc hóa các **thuộc tính numeric** bằng cách dùng **MDL-base discretization method** |
| 4 | **OneRAttributeEval** | Sử dụng một phương pháp đô độ chính xác đơn giản được kết thừa từ **OneR classifier**. Nó có thể dùng dữ liệu huấn luyện để đánh giá hoặc nó có thể áp dụng **internal cross-validation** |
| 5 | **PrincipalComponents** | Biến đổi một tập các attribute . Các attribute mới được xếp hạng dựa vào thứ tự của **eigenvalues**. **Tập con (Subset)** được chọn bằng cách chọn số lượng vừa đủ các **vector trị riêng (eigenvector)** để thỏa mãn một phương sai cho trước. |
| 6 | **ReliefFAttributeEval** | Đây là phương pháp dựa vào **instance**. Phương pháp này lấy mẫu một các ngẫu nhiên và kiểm tra các **instance gần đó (neighboring insances)** mà có cùng hoặc khác **class**. Phương pháp này hoạt động cả khi class mang giá trị rời rạc hoặc liên tục. |
| 7 | **SymmetricalUncertAttributeEval** | Đánh giá một thuộc tính bằng cách đo **symmetrical uncertainty** của nó cho **class**. |
| 8 | **WrapperSubsetEval** | Sử dụng classifier để đánh giá tập attribute và áp **dụng cross-validation** để dự đoán độ chính xác của **lược đồ học (learning scheme)** cho từng tập. |



Hình 18: Các phương pháp Search

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN PHƯƠNG PHÁP** | **MỤC ĐÍCH** |
| 1 | **BestFirst** | Thực hiện **leo đồi tham lam (greedy hill climbing)** với **quay lui (backtracking)**. Nó có thể tìm kiếm **tiến (forward)** từ một tập attribute rỗng, **lui (backward)** từ tập chứa toàn bộ attribute hoặc có thể bắt đầu từ một trạng thái cụ thể nào đó và tìm kiếm theo 2 hướng |
| 2 | **GreedyStepwise** | Tìm kiếm tham lam trong không gian các tập attribute. Nó cũng có thể tìm kiếm tới và lui. Tuy nhiên, nó không sử dụng quay lui mà dừng lại ngay khi thêm hoặc xóa đi thuộc tính tốt nhất còn lại mà làm giảm số liệu đánh giá |
| 3 | **Ranker** | Nó sắp xếp các thuộc tính bằng các sự đánh giá độc lập và phải được sử dụng kết hợp với phương pháp **đánh giá single-attribute (single-attribute evaluator)**. Phương pháp này không chỉ xếp hạng các **thuộc tính (attributes)** mà còn thực hiện chọn các thuộc tính bằng các loại bỏ những thuộc tính xếp hạng thấp |

1. *So sánh với các phương pháp chọn lọc dữ liệu trong textbook, có phương*

*pháp nào không có trong Weka hay phương pháp nào trong Weka không có*

*trong textbook?*

Phương pháp không có trong Weka là **ChiSquaredAttributeEval**

Phươnh pháp trong Weka mà không có trong textbook là  
**ClassifierAttributeEval**, **ClassifierSubsetEval**, **CorrelationAttributeEval**

1. **CHUẨN BỊ DỮ LIỆU – LÀM SẠCH DỮ LIỆU**
2. *Các* ***giá trị thiếu (Missing values)****: Liệt kê các phương pháp đã học để xử lý dữ liệu thiếu. Weka đã cài đặt những phương pháp nào? Bạn hãy chọn 1 phương pháp để xử lý giá trị thiếu trong dataset, giải thích tại sao bạn chọn phương pháp đó. Cài đặt 1 phương pháp khác mà bạn thích nếu nó không có trong Weka*

Các phương pháp để xử lý dữ liệu thiếu là:

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN PHƯƠNG PHÁP** | **MỤC ĐÍCH** |
| 1 | **Loại bỏ các dòng (Ignoring the tuple)** | Phương pháp này được sử dụng khi dòng đó không có (giả thiết rằng việc khai thác đòi hỏi phân lớp). Phương pháp này không hiệu quả trừ khi dòng đó có vài thuộc tính thiếu giá trị và nó sẽ hoạt động kém nếu số lượng giá trị bị mất của mỗi thuộc tính biến động lớn |
| 2 | **Thêm bằng cách thủ công (Fill in the missing value manually)** | Phương pháp này nhìn chung thì tốn thời gian và có thể không khả thi nếu dataset lớn với nhiều giá trị thiếu |
| 3 | **Sử dụng một hằng số để thay thế giá trị thiếu (Use a global constant to fill in the missing value)** | Thay thế tất cả các giá trị thiếu của thuộc tính bằng một hằng số giống nhau như **“Unknown”** hoặc **-∞**. Nếu giá trị thiếu được thay thế bằng **“Unknown”** thì chương trình khai thác sẽ nhầm lẫn, tưởng rằng nó đang làm việc với một khái niệm đặc biệt. Do đó, phương pháp này dễ nhưng không hoàn hảo. |
| 4 | **Use the attribute mean to fill in the missing value** | Sử dụng trung bình của attribute để thay thế dữ liệu thiếu |
| 5 | **Use the attribute mean for all samples belonging to the same class as given tuple** | Sử dụng trung bình của attribute cho tất cả các mẫu cùng thuộc về một lớp giống như tuple được cho |
| 6 | **Sử dụng giá trị có khả năng nhất để thay thế giá trị thiếu (Use the most probable value to fill in the missing value)** | Có thể được xác định bằng hồi quy, cây quyết định |

Weka đã cài đặt phương pháp: (3), (4)

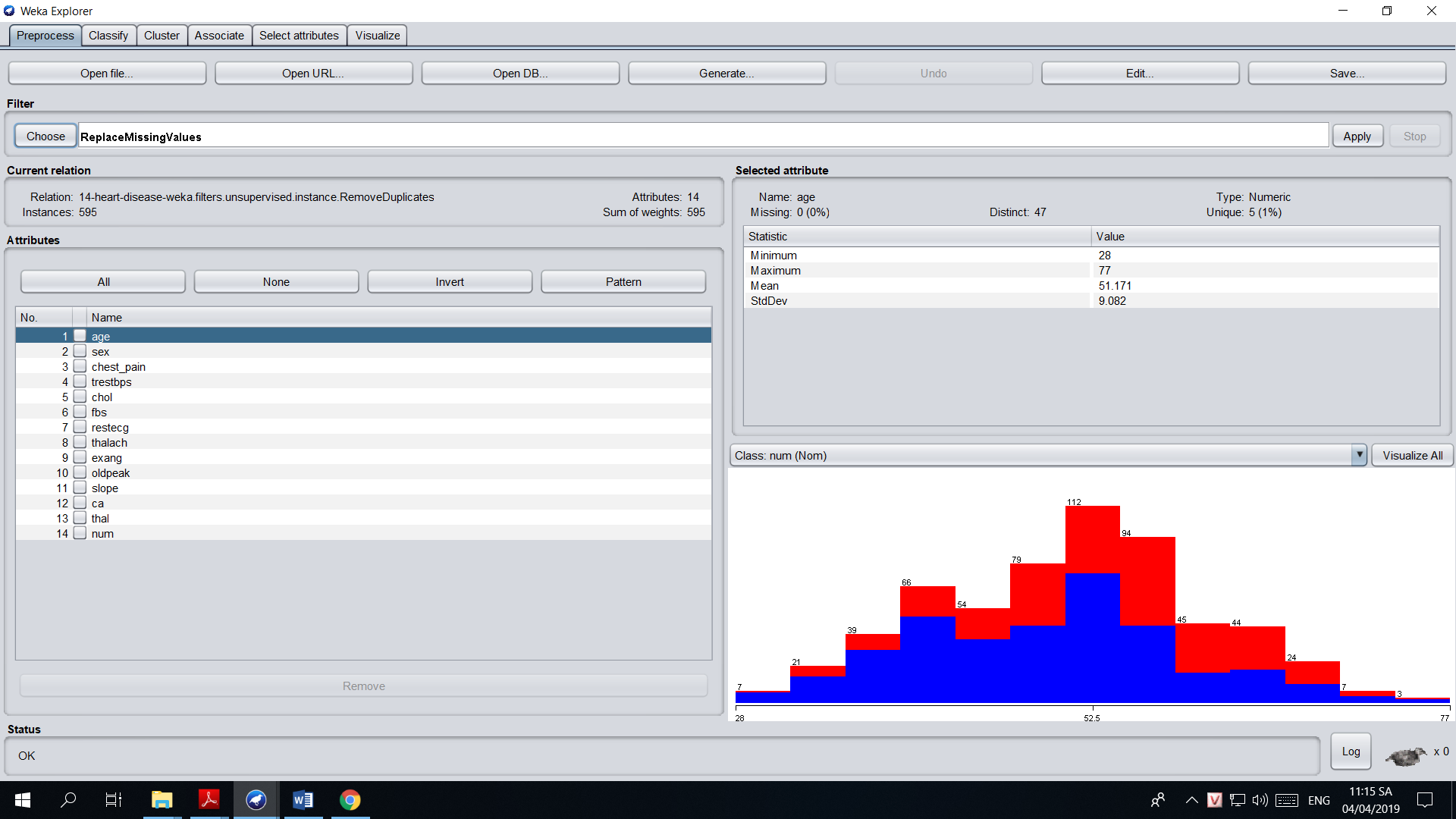
1. ***Dữ liệu nhiễu (Noisy data)****: Liệt kê các phương pháp đã học để loại bỏ các dữ liệu nhiễu, Weka đã cài đặt những phương pháp nào?*

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN PHƯƠNG PHÁP** | **MỤC ĐÍCH** |
| 1 | **Binning** | Phương pháp này làm trơn dữ liệu đã được sắp xếp bằng cách dựa vào các giá trị xung quanh. Do đó, phương pháp này là làm trơn cục bộ |
| 2 | **Regression** | Data có thể được làm trơn bằng cách xây dựng một hàm hồi quy cho nó. Hồi quy tuyến tính tìm đường thẳng tốt nhất để biểu diễn 2 attribute, do đó, một thuộc tính sẽ được suy ra từ thuộc tính còn lại. **Multiple linear regression** là phiên bản mở rộng của **linear regression**, có nhiều hơn 2 attribute tham gia và dữ liệu được biểu diễn bằng một mặt phẳng đa chiều |
| 3 | **Clustering** | Điểm **outlier** có thể được phát hiện bằng phương pháp phân cụm. Những giá trị nào mà nằm ngoài tất cả các cụm thì được xem là **outlier** |

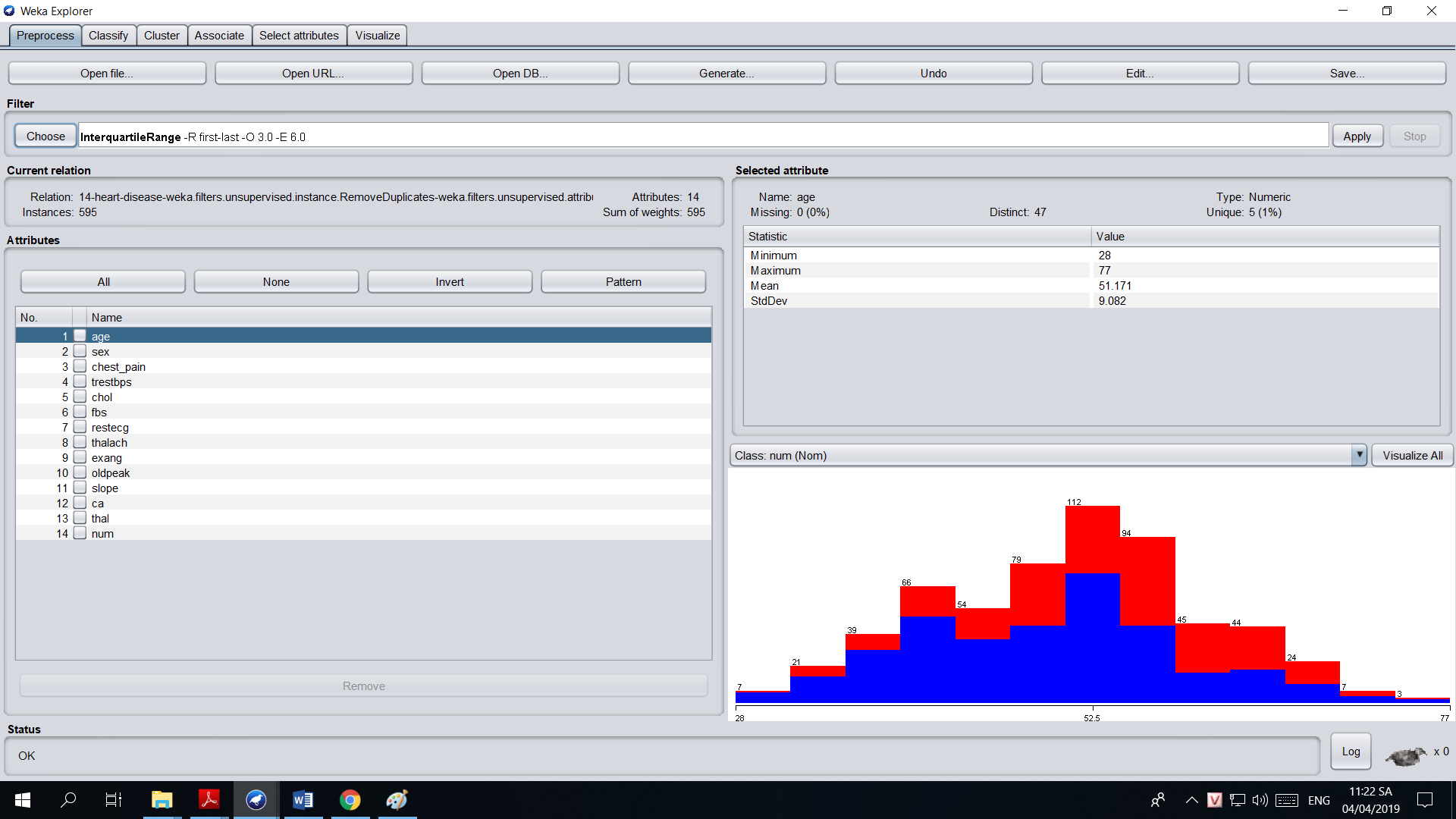
1. *Dò tìm* ***dữ liệu tạp (Outlier detection)****: Liệt kê các phương pháp đã học để dò tìm dữ liệu tạp. Bạn dò tìm dữ liệu tạp bằng Weka như thế nào? Có dữ liệu tạp trong dataset đã cho hay không? Nếu có, liệt kê một số dữ liệu tạp.*

Các phương pháp dò tìm dữ liệu tạp là **Clustering**

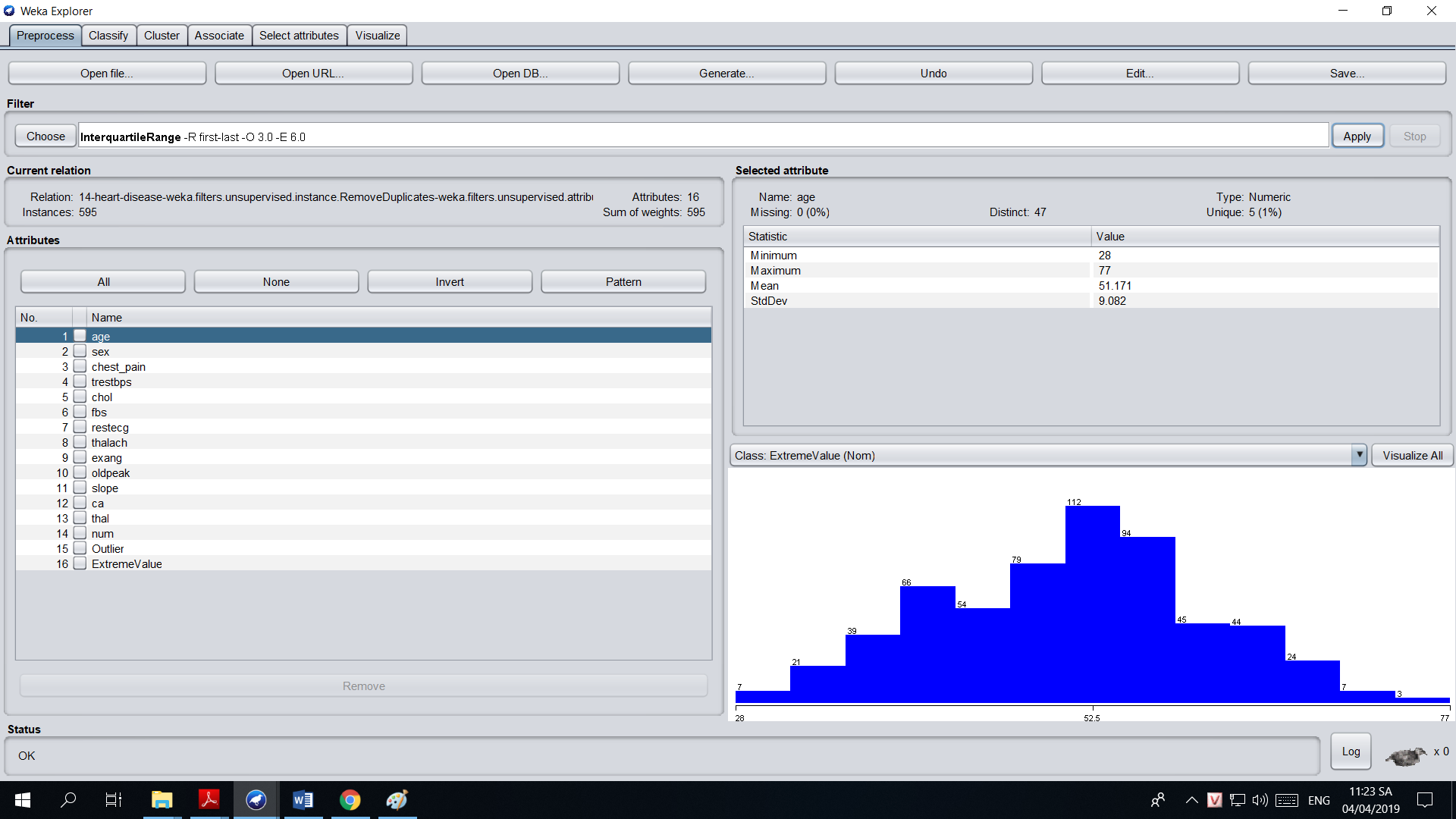
Em dò tìm dữ liệu bằng cách **ReplaceMissingValues** sau đó sử dụng **InterquartileRange** để tìm ra **Outliers** và **ExtremeValues**. Sau đó, em dùng **RemoveWithValues** để loại bỏ các dòng dữ liệu dựa vào thuộc tính của dòng đó.



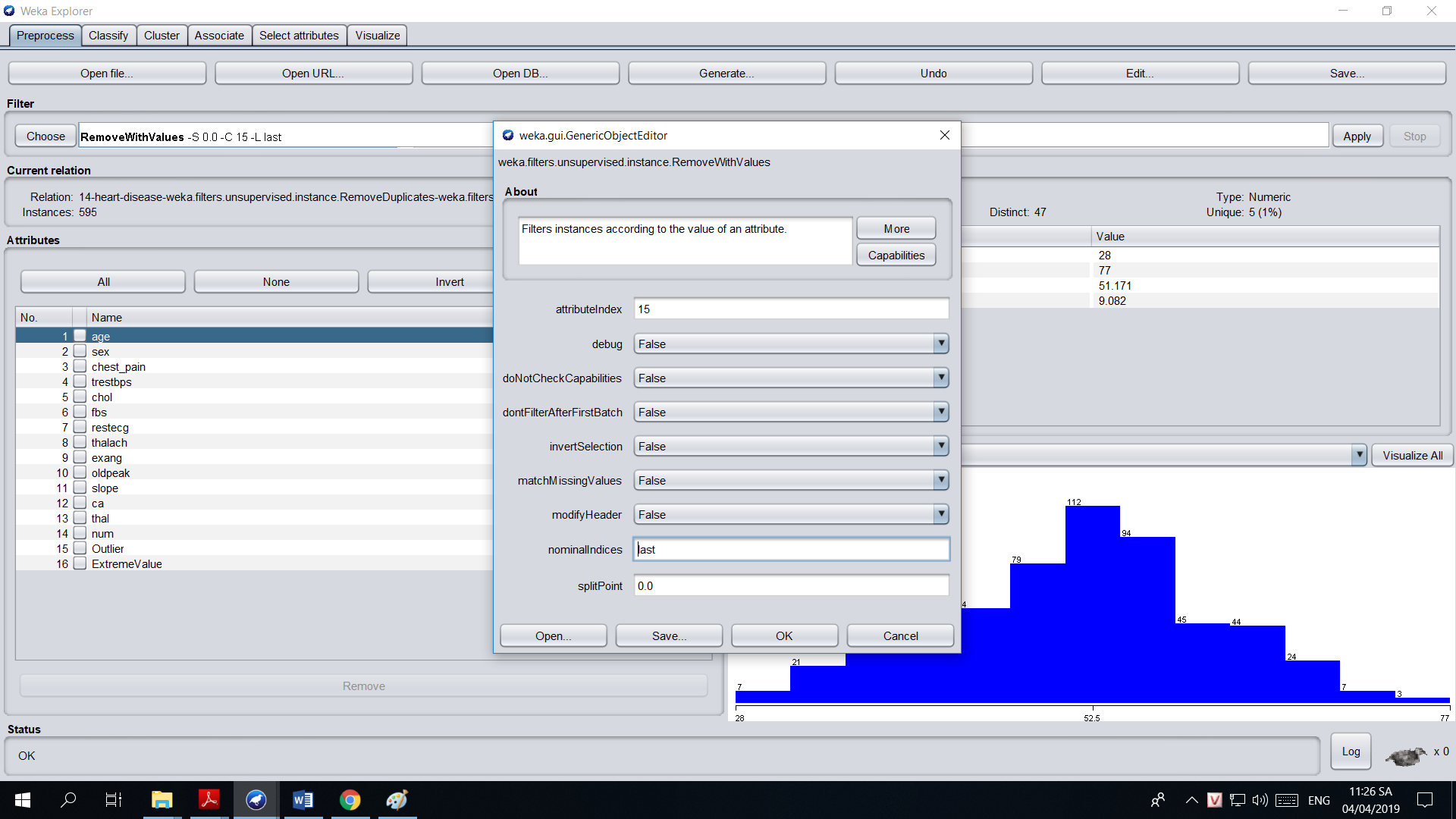
Hình 19: Sử dụng filter ReplaceMissingValues



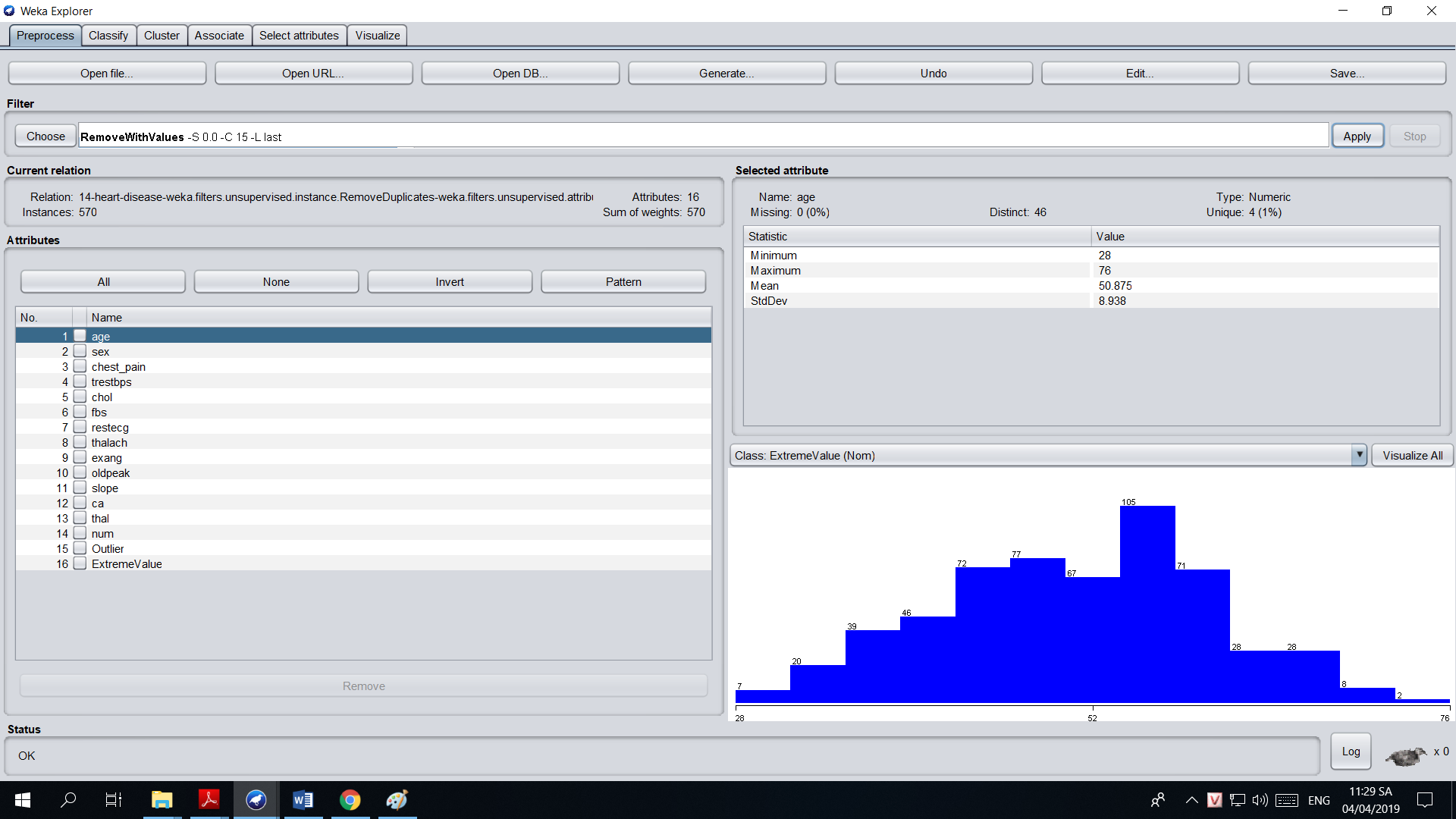
Hình 20: Sử dụng filter InterquartileRange



Hình 21: Tạo ra 2 attribute là Outlier và ExtremeValue

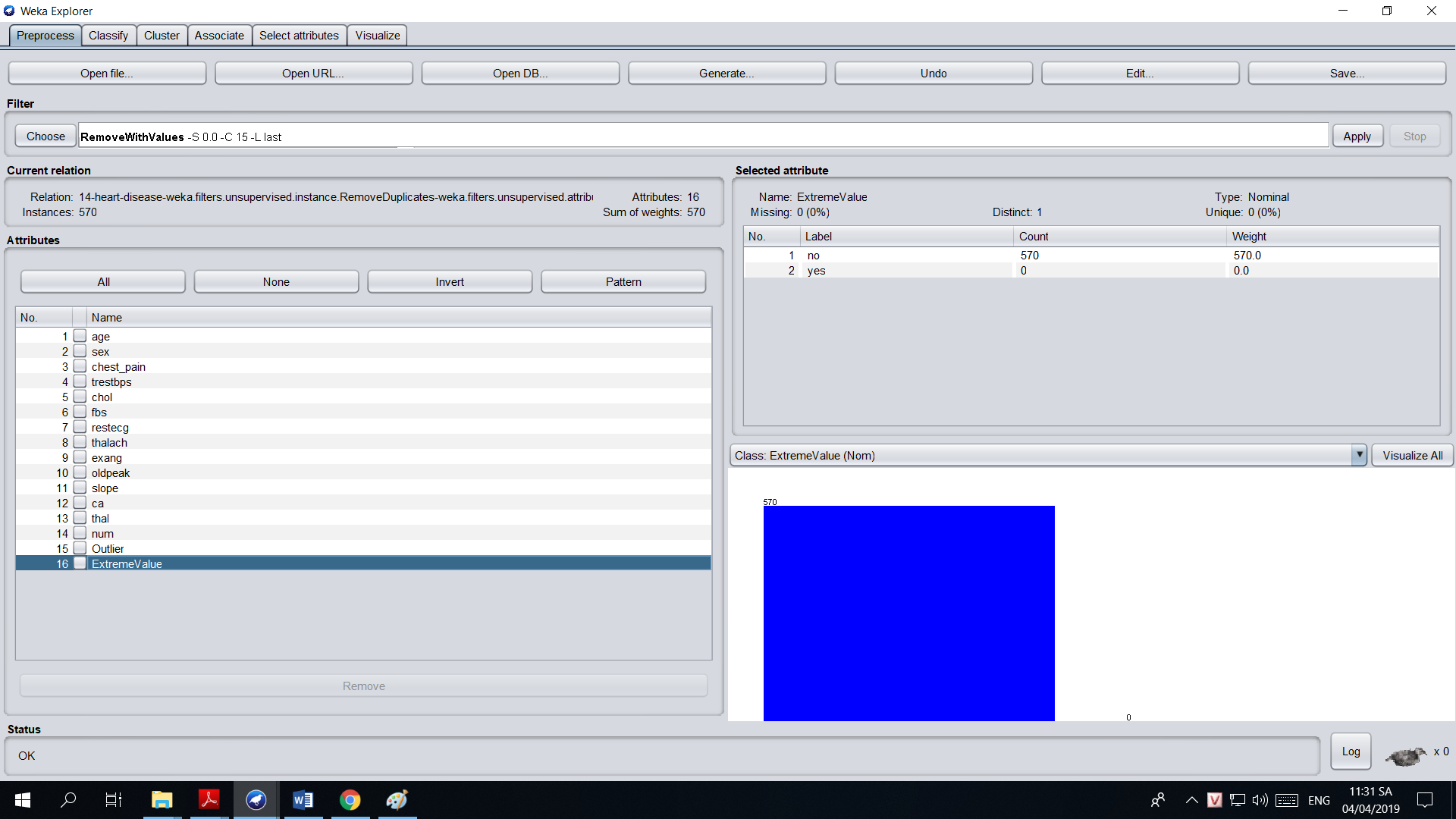


Hình 22: Sử dụng filter RemoveWithValues với tham số là 15 và last



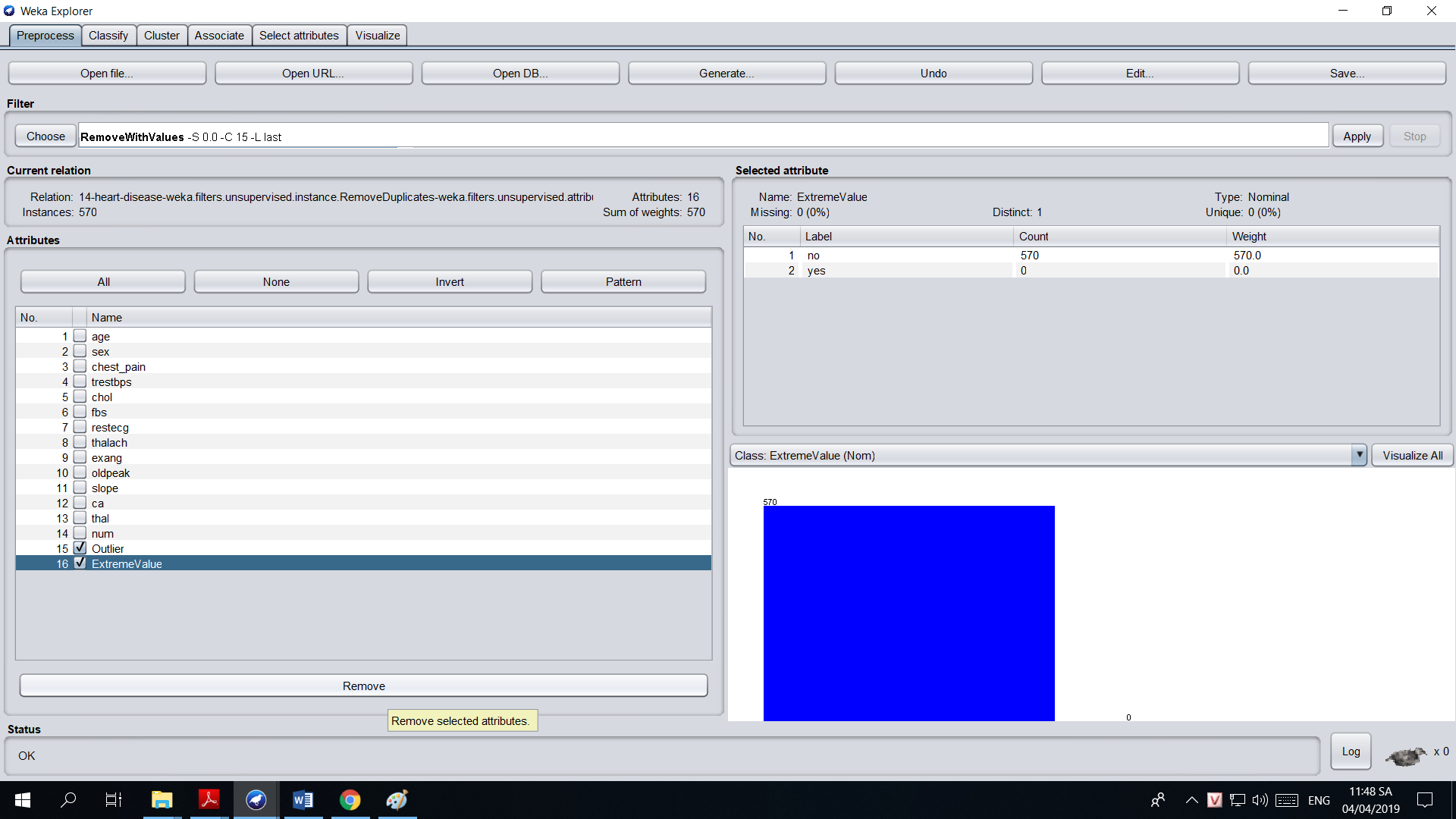
Hình 23: Từ 595 instances chúng ta còn 570

Vì **ExtremeValue** không có **instances** mang giá trị **yes** nên ta không cần xóa các dòng mang giá trị **yes**



Hình 24: Thuộc tính ExtremeValue

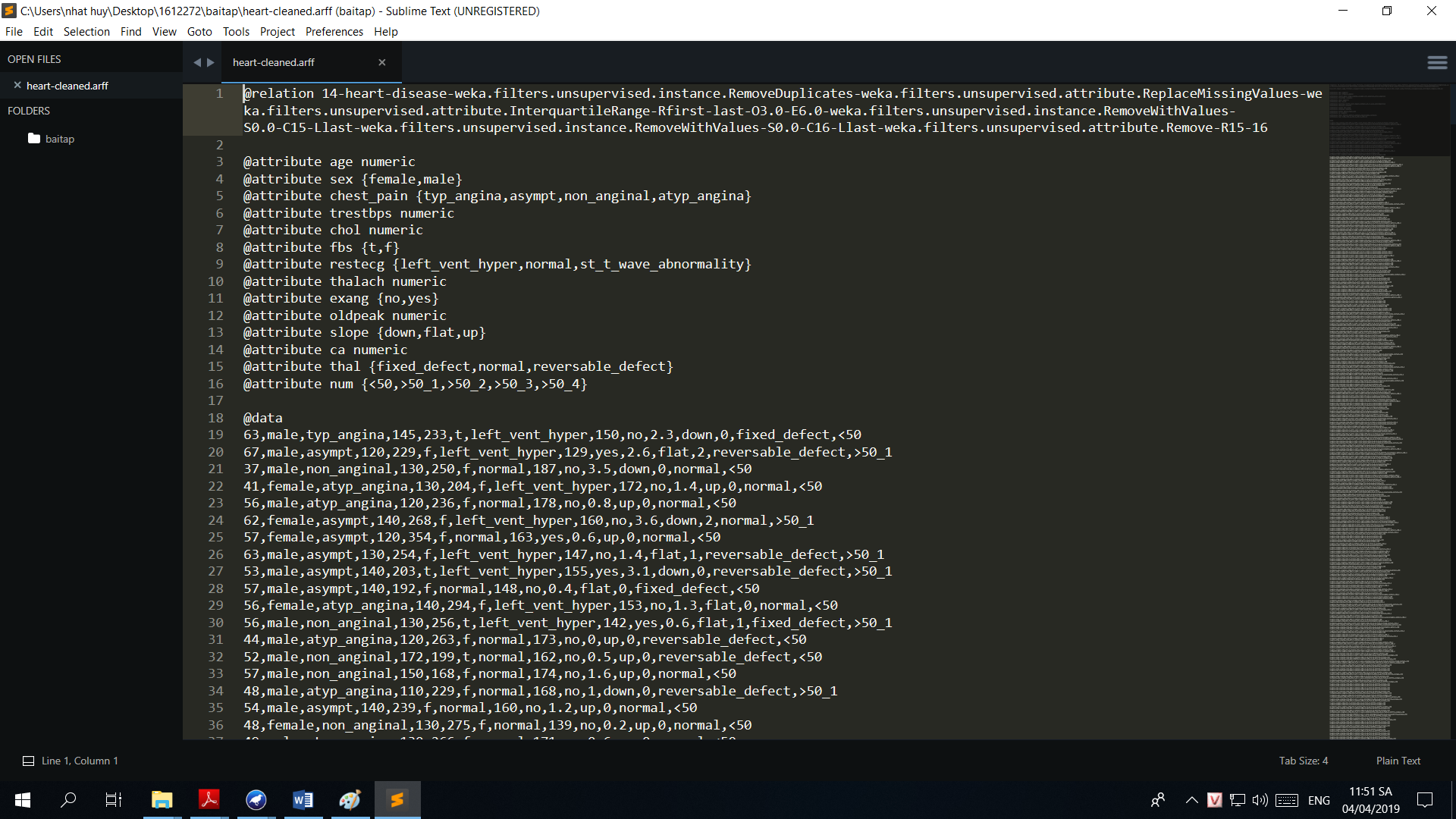
Ta sẽ xóa 2 thuộc tính **Outlier** và **ExtremeValue** khỏi dataset và lưu lại thành file **heart-cleaned.arff**



Hình 25: Nhấn Remove để loại bỏ 2 thuộc tính này và Save

1. *Lưu dataset đã làm sạch vào file* ***heart-cleaned.arff*** *và dán vào bài làm 1 ảnh*

*chụp cho thấy ít nhất 10 dòng của dữ liệu với tất cả các cột.*

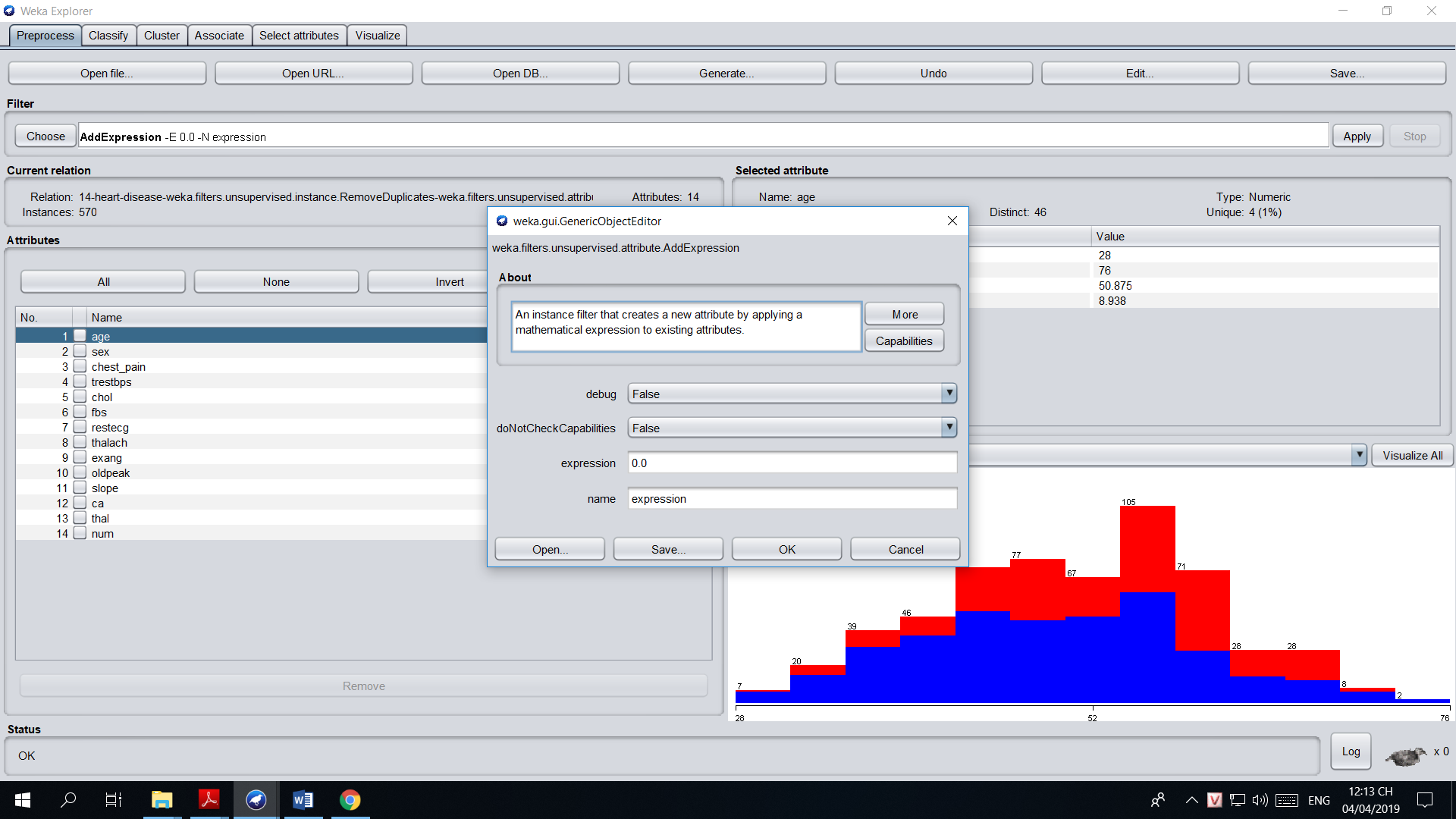
****

Hình 26: Dữ liệu sau khi làm sạch

1. **CHUẨN BỊ DỮ LIỆU – CHUYỂN ĐỔI DỮ LIỆU**
2. ***Xây dựng thuộc tính – Attribute construction****: ví dụ, thêm một thuộc tính là*

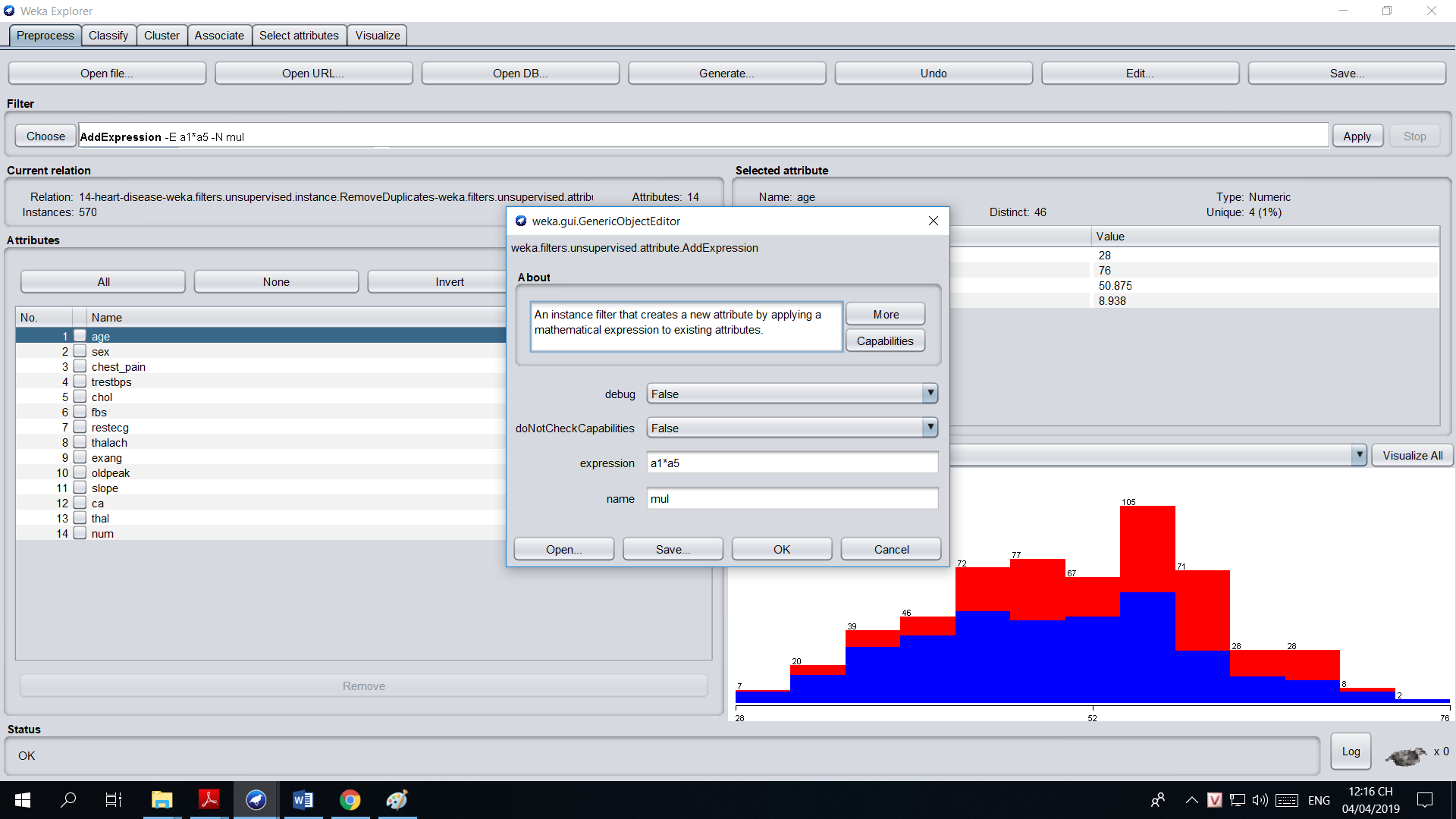
*tổng của 2 thuộc tính khác. Bộ lọc nào của Weka cho phép làm điều này?*

Trong Weka, chúng ta sẽ sử dụng bộ lọc **AddExpression** để **xây dựng thuộc tính (Attribute Constructor)**

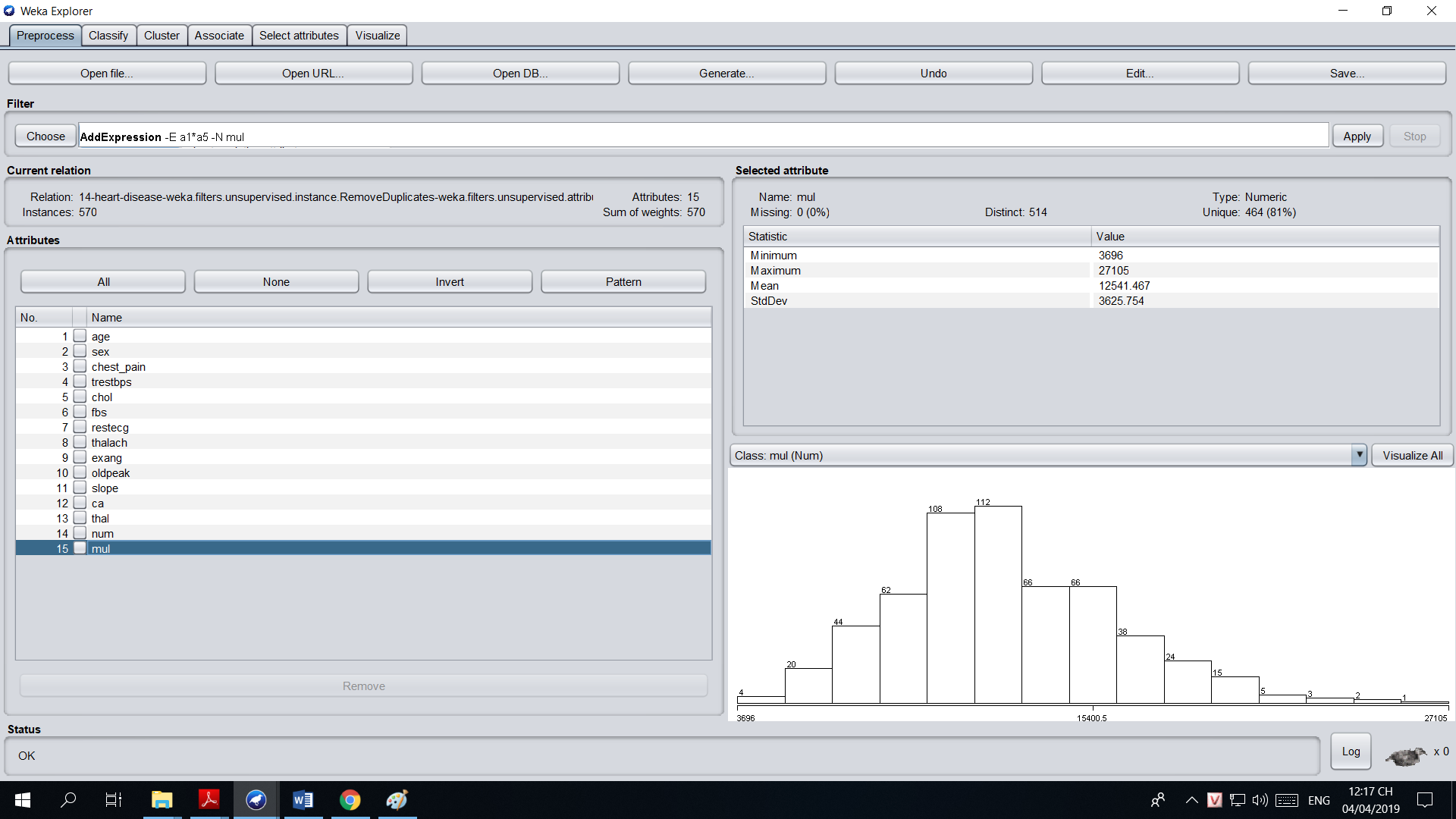


Hình 27: Bộ lọc AddExpression

Ta sẽ gõ công thức vào mục **expression** với attribute có tiền tố là **“a”** và chỉ số của nó trong dataset



Hình 28: Tạo thuộc tính mul bằng cách nhân a1 và a5



Hình 29: Attribute mới được tạo ra có tên là mul

1. ***Chuẩn hóa – Normalize*** *một thuộc tính. Bộ lọc nào của Weka cho phép làm*

*điều này? Bộ lọc đó có thể* ***chuẩn hóa Min-max*** *không,* ***chuẩn hóa Z-score*** *hay*

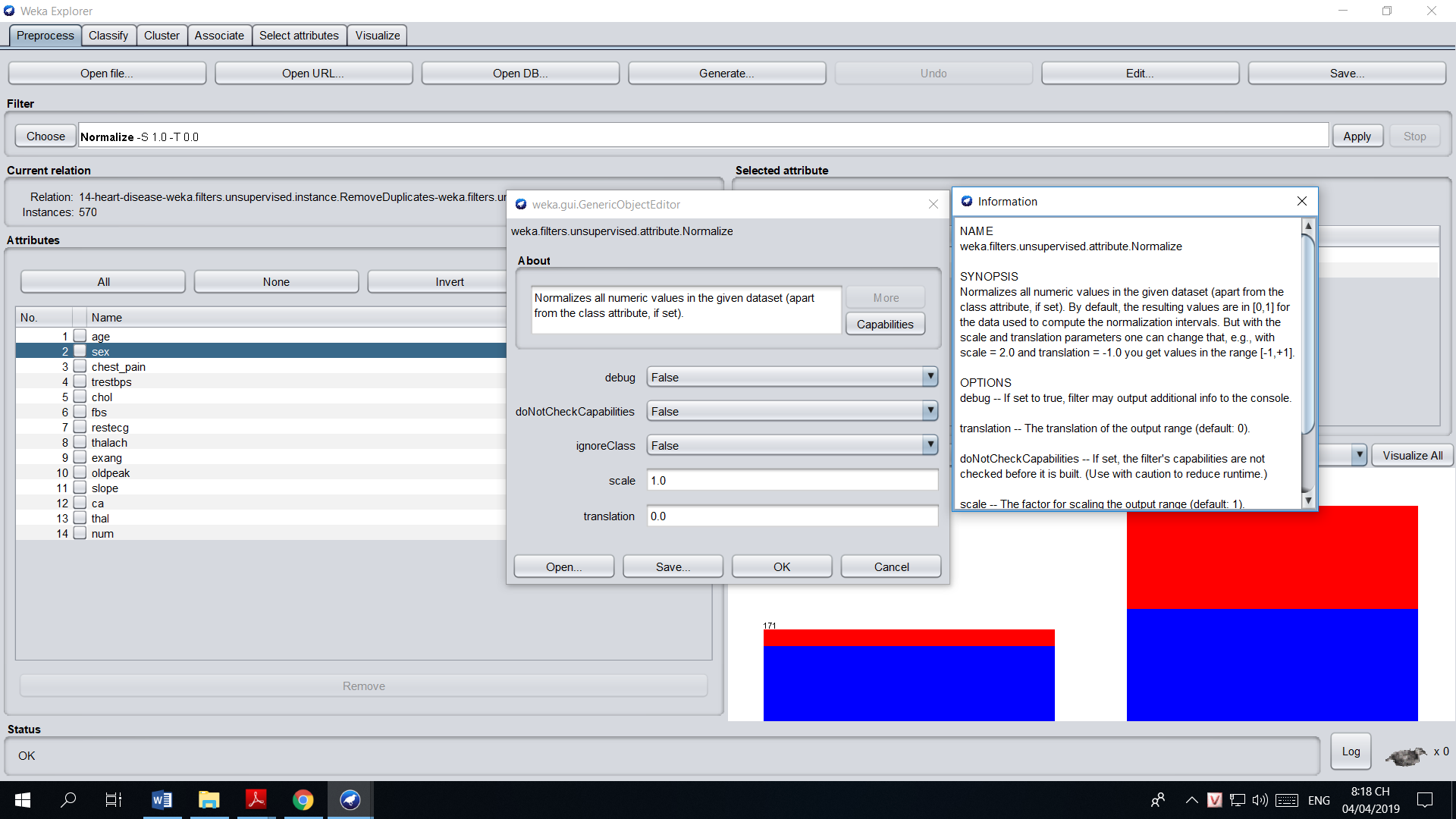
***chuẩn hóa thập phân*** *hay không? Cho biết cụ thể cách thức thực hiện những*

*chuẩn hóa này trong Weka.*

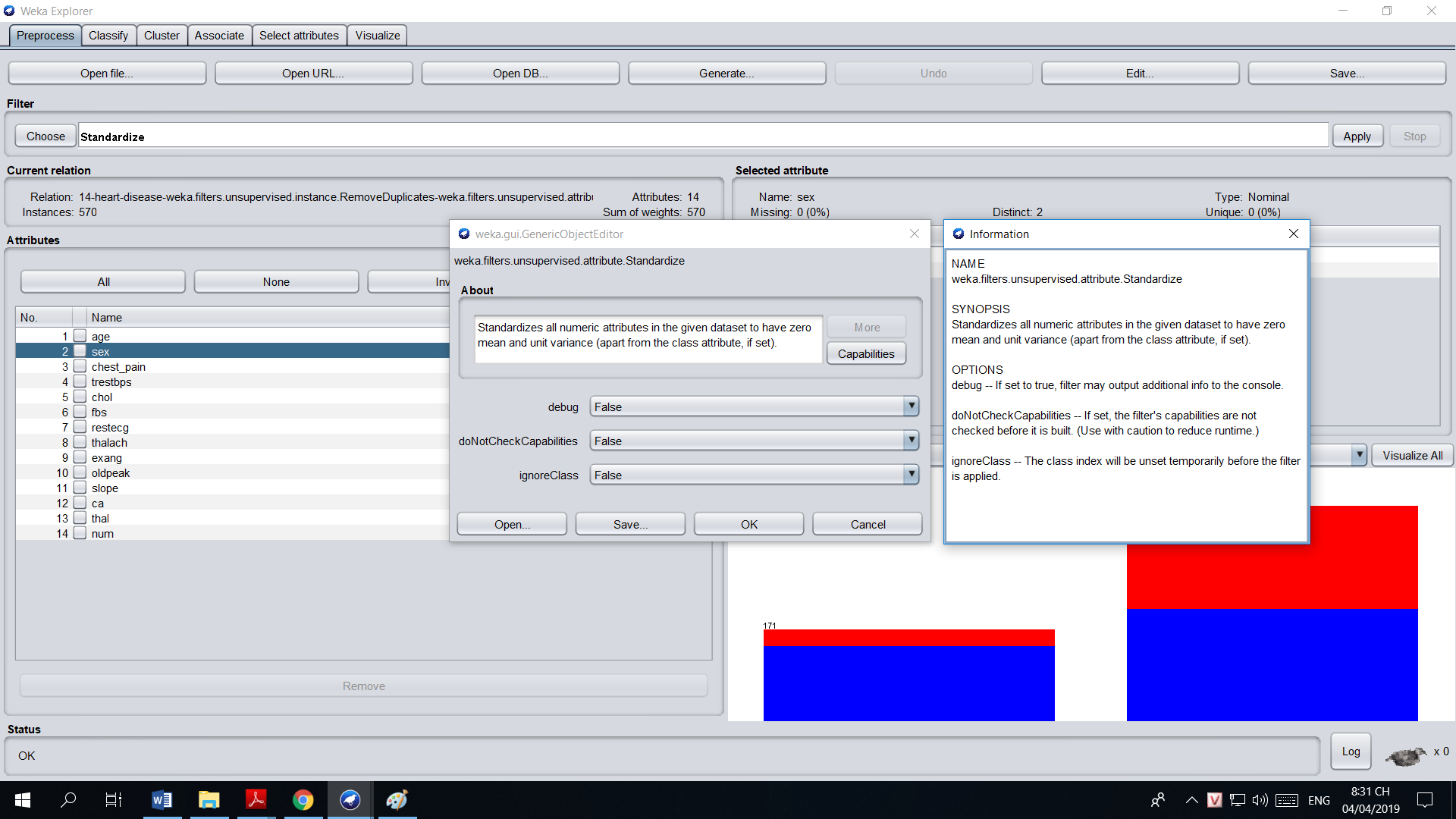
**Chuẩn hóa (Normalize)** một thuộc tính là chúng ta biến đổi bằng cách **scailing** dữ liệu của chúng sao cho giá trị nằm trong một khoảng nhỏ cụ thể như là **[0.0 ; 1.0]**. Có rất nhiều cách chuẩn hóa nhưng ta quan tâm nhất 3 loại sau:

|  |  |  |
| --- | --- | --- |
| **STT** | **TÊN PHƯƠNG PHÁP** | **MỤC ĐÍCH** |
| 1 | **Min-max normalization** | Sử dụng phép biến đổi tuyến tính dữ liệu gốc. Giả sử, chúng ta có thuộc tính **A** với giá trị nằm trong khoảng **[minA; maxA]**. Ta muốn chuẩn hóa một giá trị **v** của **A** thành **v’** thuộc đoạn **[new\_minA; new\_maxA]** bằng cách sau:  Phương pháp này bảo toàn mối quan hệ giữa các giá trị trong dữ liệu gốc. Tuy nhiên, phương pháp này gặp lỗi cho các dữ liệu trong tương lai mà nằm ngoài khoảng giá trị của **A** |
| 2 | **z-score normilization** | Giá trị của thuộc tính **A** được chuẩn hóa dựa **vào trung bình (mean)** và **độ lệch chuẩn (standard deviation)** của **A**. Giá trị **v** của **A** được chuẩn hóa thành **v’** bằng cách sau: , với và lần lượt là **trung bình** và **độ lệch chuẩn** của **A**. Phương pháp này tốt khi mà giá trị **min** và **max** của **A** là không xác định được hoặc dữ liệu nhiễu gây sai cho phương pháp **min-max normalization** |
| 3 | **Chuẩn hóa thập phân (Normalization by decimal scailing)** | Chuẩn hóa bằng cách di chuyển chấm thập phân của các giá trị thuộc tính **A**. Số chấm thập phân di chuyển sẽ phụ thuộc vào giá trị tuyệt đối lớn nhất của **A**. Giá trị **v** của **A** được chuẩn hóa thành **v’** bằng cách sau: , với **j** là số nguyên nhỏ nhất để **Max(|v’|) < 1** |

Trong Weka, chúng ta được cung cấp 2 bộ lọc để chuẩn hóa là **min-max normalization** và **z-score normalization**



Hình 30: Min-max normalization

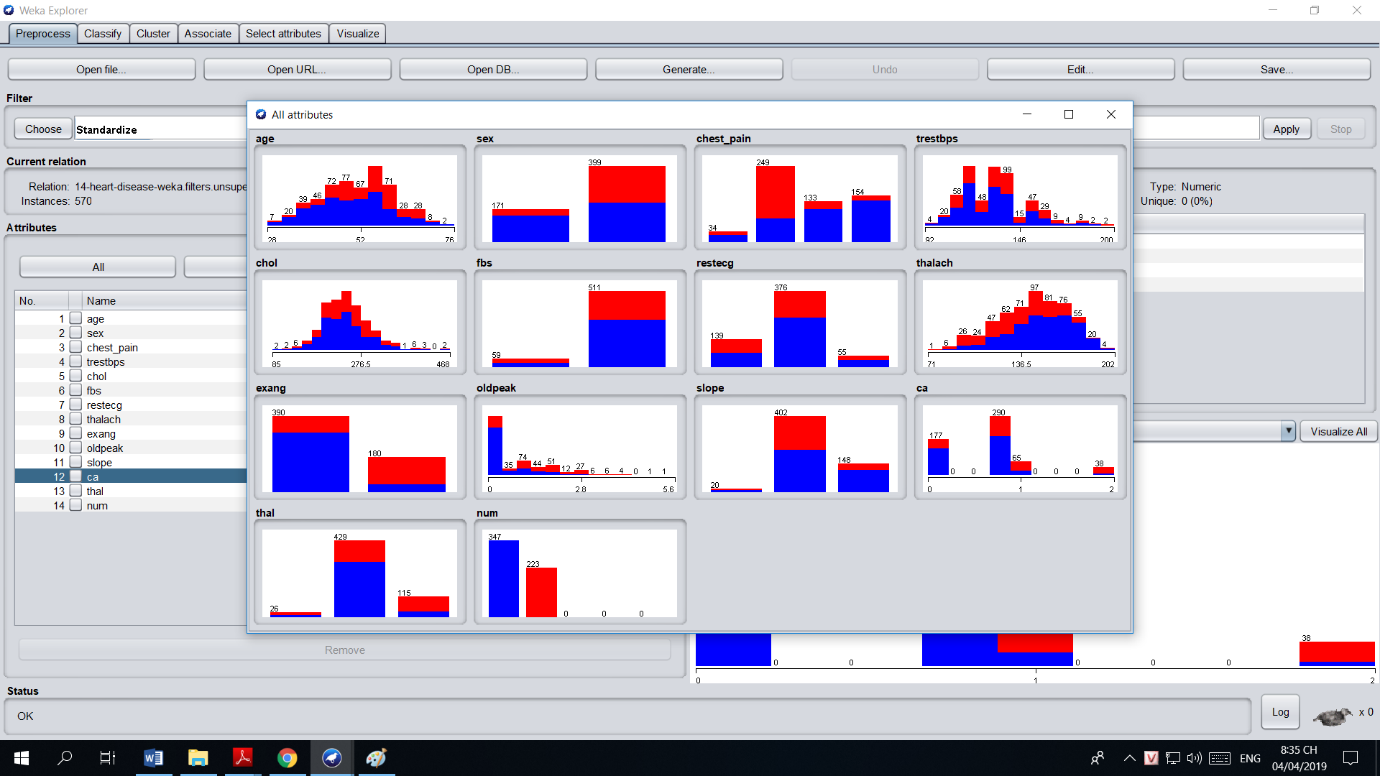


Hình 31: z-score normalization

1. *Chọn 1 phương pháp và tiến hành chuẩn hóa tất cả các thuộc tính là số thực,*

*giải thích sự lựa chọn của bạn.*

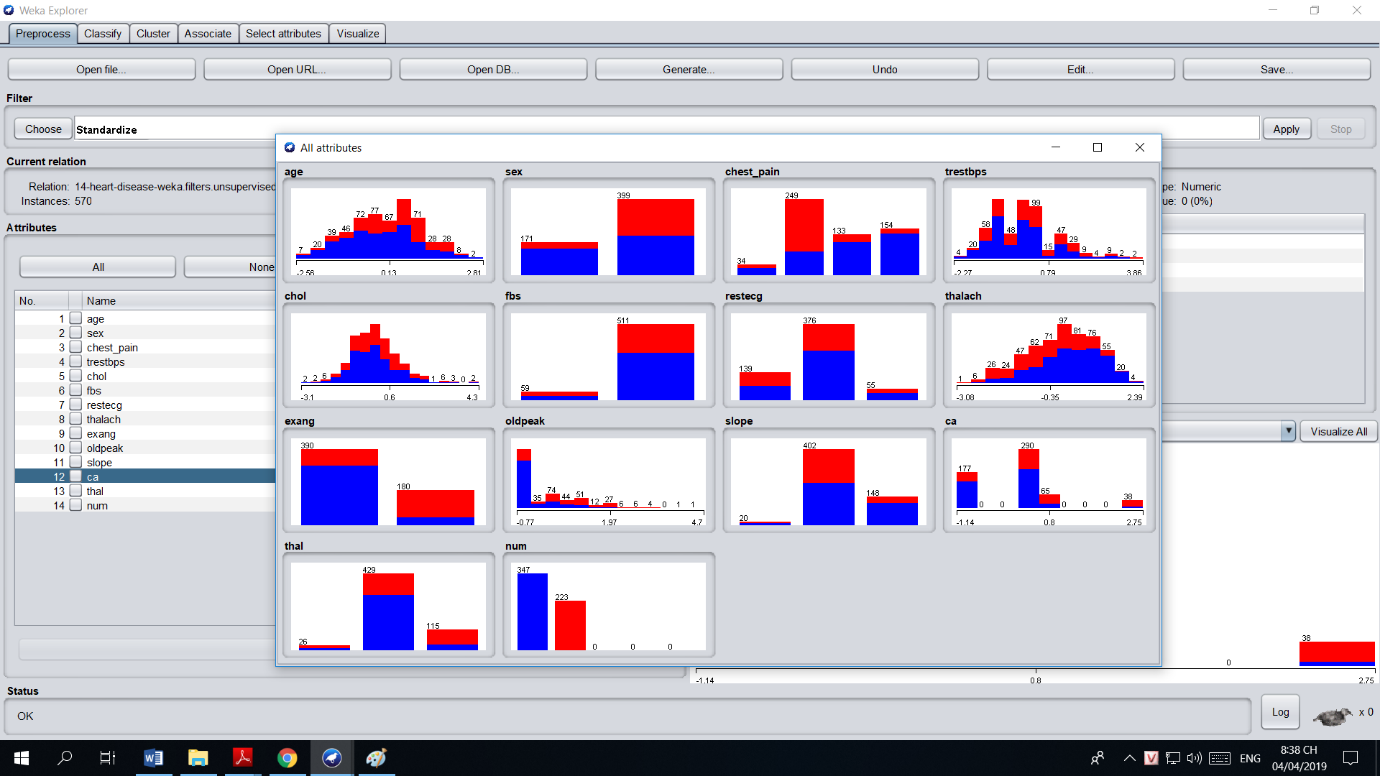
Ta chọn phương pháp **z-score normalization** để chuẩn hóa vì các thuộc tính là số thực trong dataset đa phần có phân phối giống với **Gauss** trừ ca nên sử dụng phương pháp này sẽ phù hợp hơn



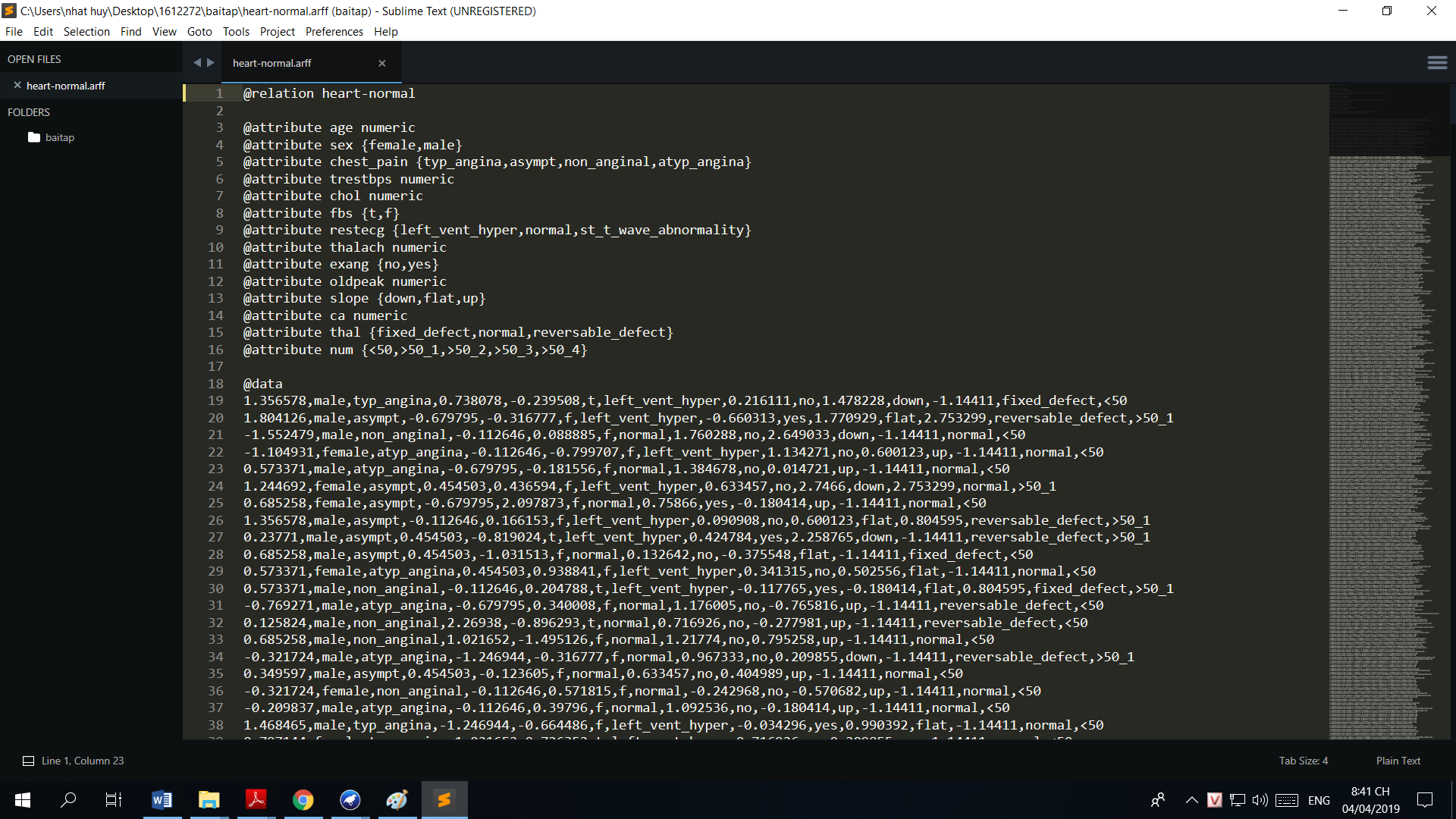
Hình 32: Phân phối của thuộc tính số có phân phối giống Gauss

1. *Lưu dataset đã chuẩn hóa vào file* ***heart-normal.arff*** *và chụp ảnh màn hình*

*cho thấy ít nhất 10 dòng dữ liệu với tất cả các cột.*



Hình 33: Kết quả sau khi dùng chuẩn hóa z-score



Hình 34: heart-normal.arff

1. **CHUẨN BỊ DỮ LIỆU – RÚT GỌN DỮ LIỆU**

*Các cơ sở dữ liệu thường rất lớn, không thể thao tác trực tiếp được. Các kỹ thuật*

*rút gọn dữ liệu được áp dụng để tiền xử lý dữ liệu. Trong* ***tab Preprocess****, bên*

*cạnh việc chọn lọc thuộc tính, một phương pháp để rút gọn dữ liệu là chọc lọc các dòng trong một dataset, hay còn gọi là* ***lấy mẫu (sampling)****. Làm cách nào để lấy mẫu với các bộ lọc của Weka? Nó có thể thực hiện 2 phương pháp chính là:* ***Simple Random Sample Without Replacement,*** *và* ***Simple Random Sample***

***With Replacement*** *hay không?*

**Rút gọn dữ liệu (Data reduction)** là kỹ thuật được áp dụng để tạo ra một dataset với cách trình bày giản lược và nhỏ hơn rất nhiều so với dữ liệu gốc nhưng vẫn giữ được sự toàn vẹn của dữ liệu gốc. Khi đó, làm việc trên dữ liệu giản lược sẽ đơn giản hơn rất nhiều. Các chiến lược cho rút gọn dữ liệu gồm:

* Data cube aggregation
* Attribute subset selection

Stepwise forward selection

Stepwise backward elimination

Combination of forward selection and backward elimination

Decision tree induction

* Dimensionality reduction

Wavelet transforms

Principal Components Analysis

* Numerosity reduction

Regression and log linear models

Histograms

Clustering

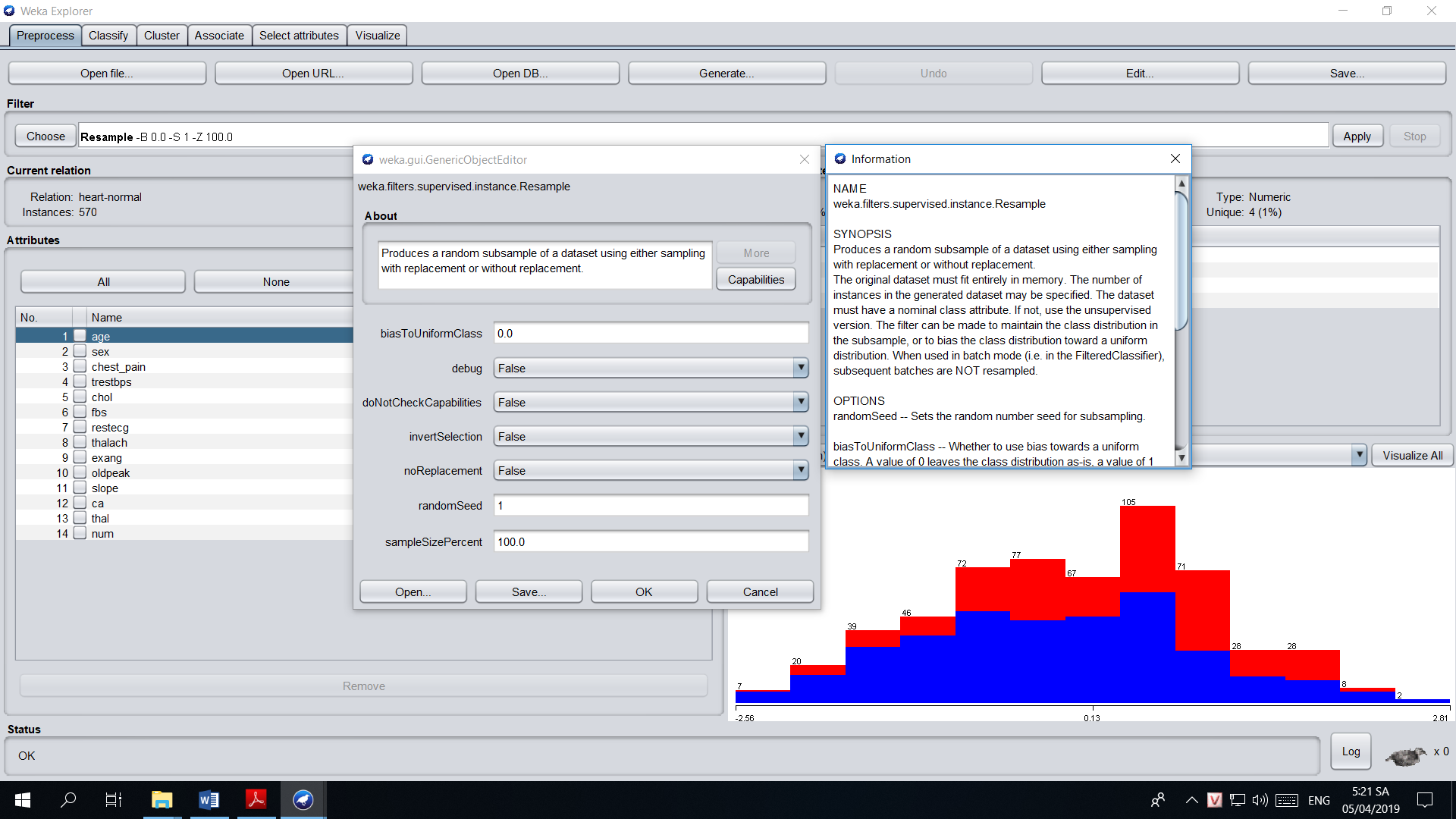
Sampling

* Discretization and concept hierarchy generation

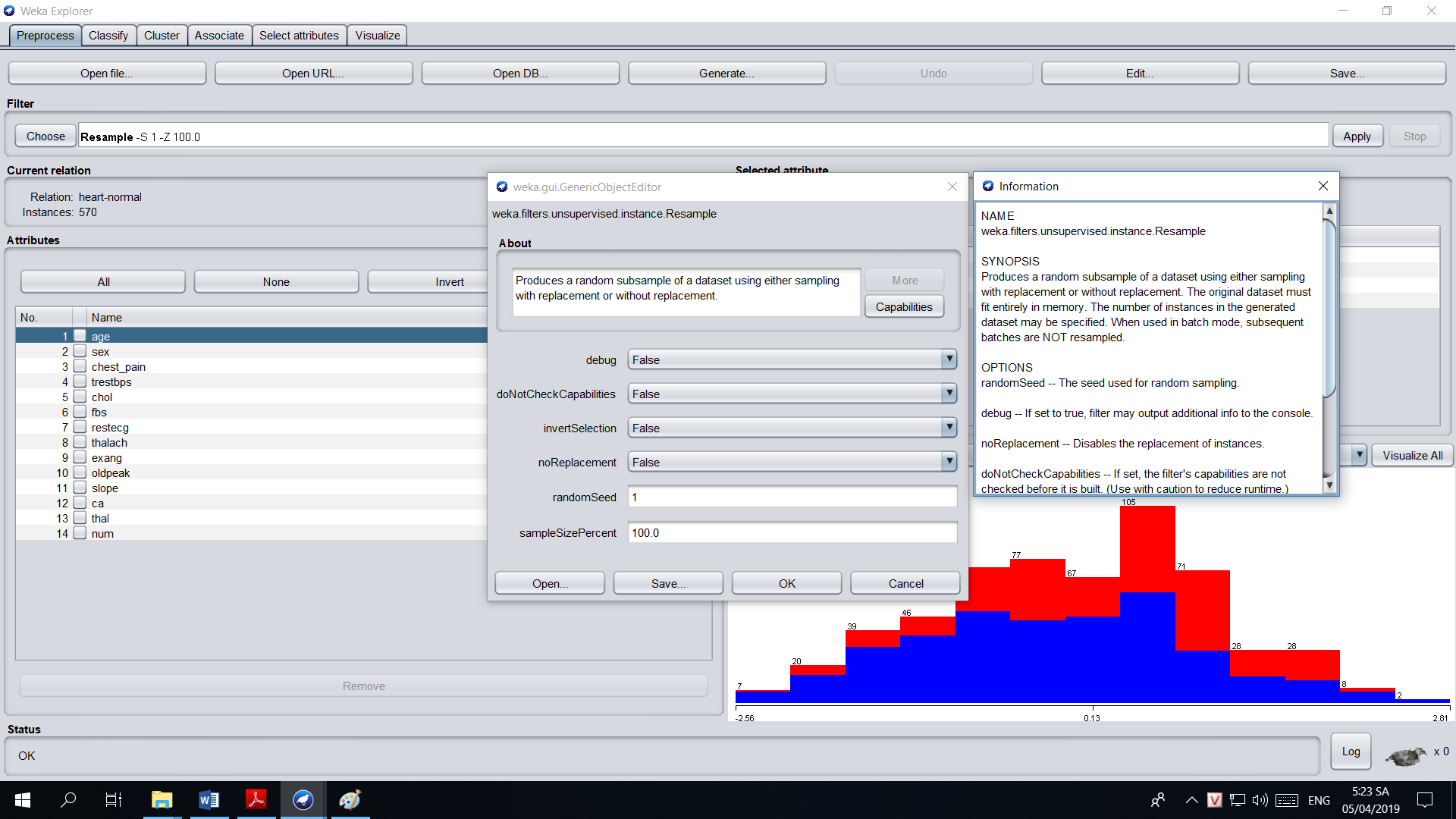
**Lấy mẫu (Sampling)** là một phương pháp thuộc nhóm **Numerosity reduction**. Lấy mẫu được sử dụng vì nó cho phép một dataset lớn được biểu diễn bởi một mẫu ngẫu nhiên nhỏ hơn dataset gốc.

* Lấy mẫu ngẫu nhiên không hoàn lại (Simple random sample without replacement): lấy mẫu bằng cách lấy ngẫu nhiên s dòng dữ liệu trong dataset, với xác suất lấy mỗi dòng là như nhau và bằng với N là số dòng dữ liệu trong dataset
* Lấy mẫu ngẫu nhiên có hoàn lại (Simple random sample with replacement): cũng lấy ngẫu nhiên tuy nhiên sau mỗi lần lấy 1 dòng dữ liệu thì dòng đó không bị loại bỏ cho lần lấy sau. Do đó, chúng ta sẽ có thể lấy ngẫu nhiên lại dòng dữ liệu này
* Cluster sample: Nếu các dữ liệu trong dataset được nhóm thành M cluster riêng lẻ nhau thì chúng ta có thể thực hiện phương pháp lấy ngẫu nhiên s cluster với s < M
* Stratified sample: Nếu dataset được chia làm các phần riêng lẻ nhau gọi là tầng (strata) thì chúng ta có thể tạo ra một mẫu phân tầng bằng cách lấy mẫu ngẫu nhiên tại mỗi tầng

Chúng ta có thể lấy mẫu bằng cách sử dụng **bộ lọc Resample**. Có 2 bộ lọc **là bộ lọc Resample** trong nhóm **Unsupervised** và **Supervised**



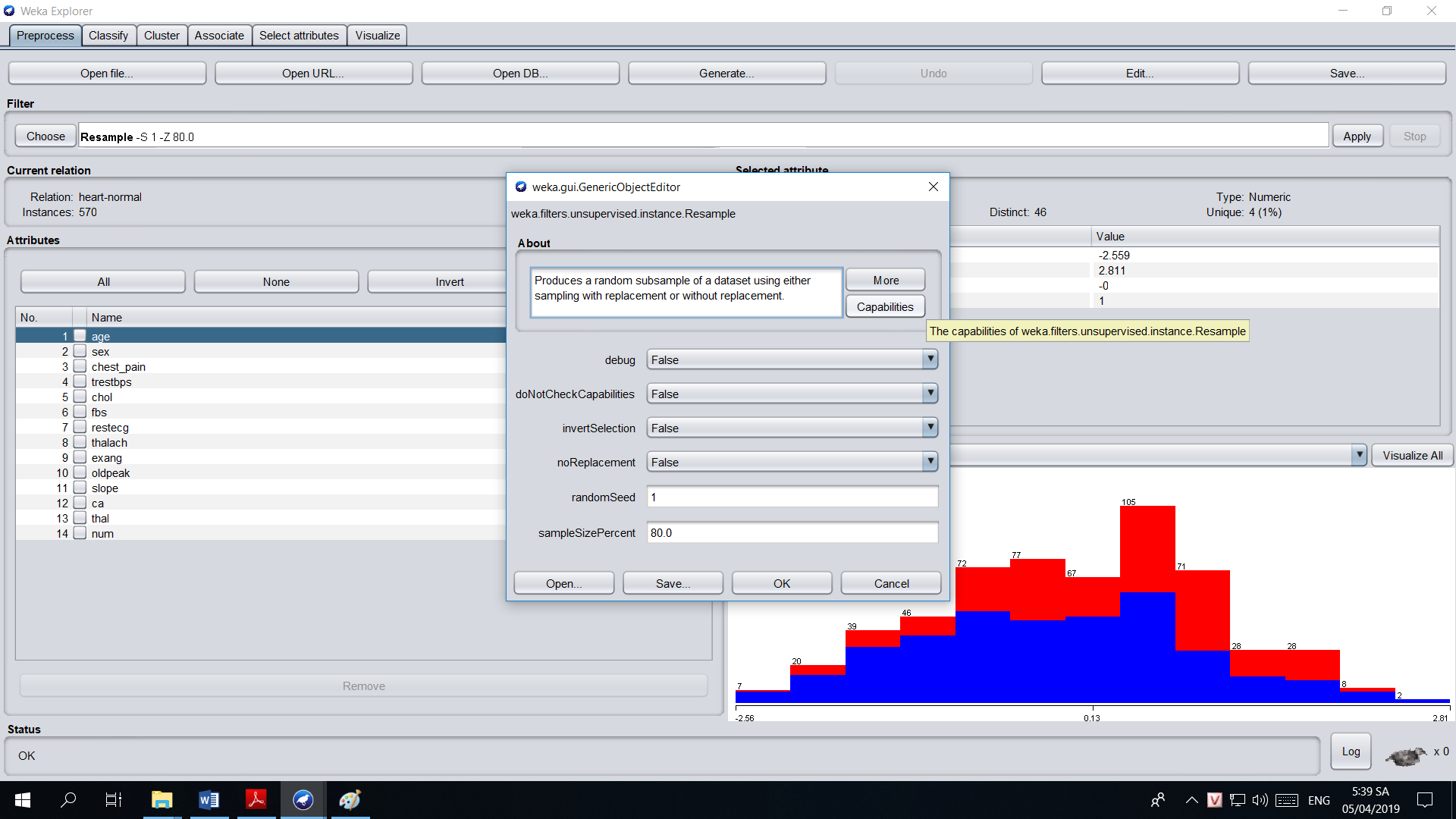
Hình 35: Bộ lọc Resample trong nhóm Supervised



Hình 36: Bộ lọc Resample trong nhóm Unsupervised

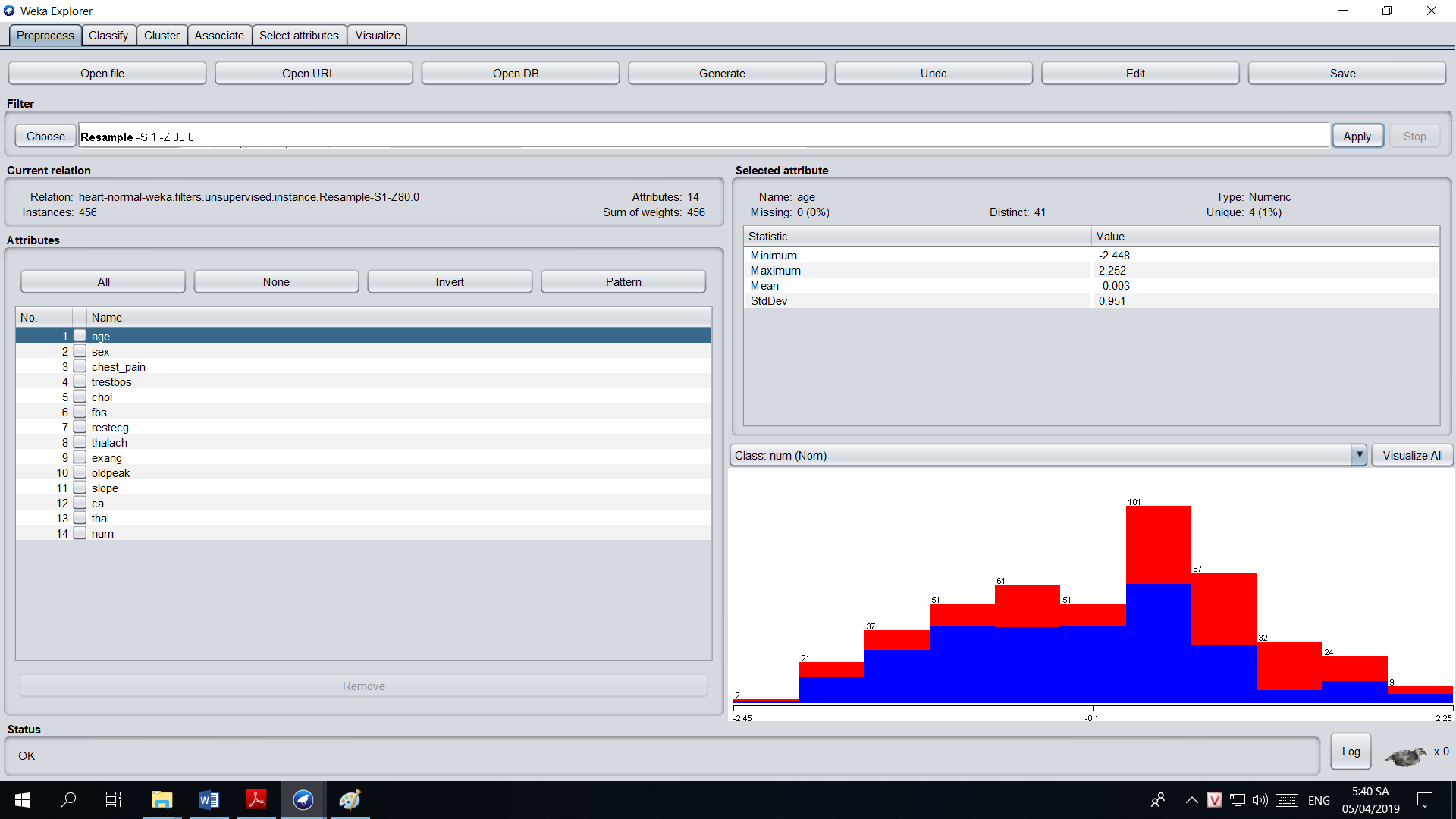
**Bộ lọc Resample** trong **Supervised** chỉ làm việc với **nominal class** hoặc **binary class** và cả 2 bộ lọc này đều cho tùy chọn **Simple random sample with replacement** và **Simple random sample without replacement**

Ta sẽ làm thử với bộ lọc **Resample** trong **Supervised**



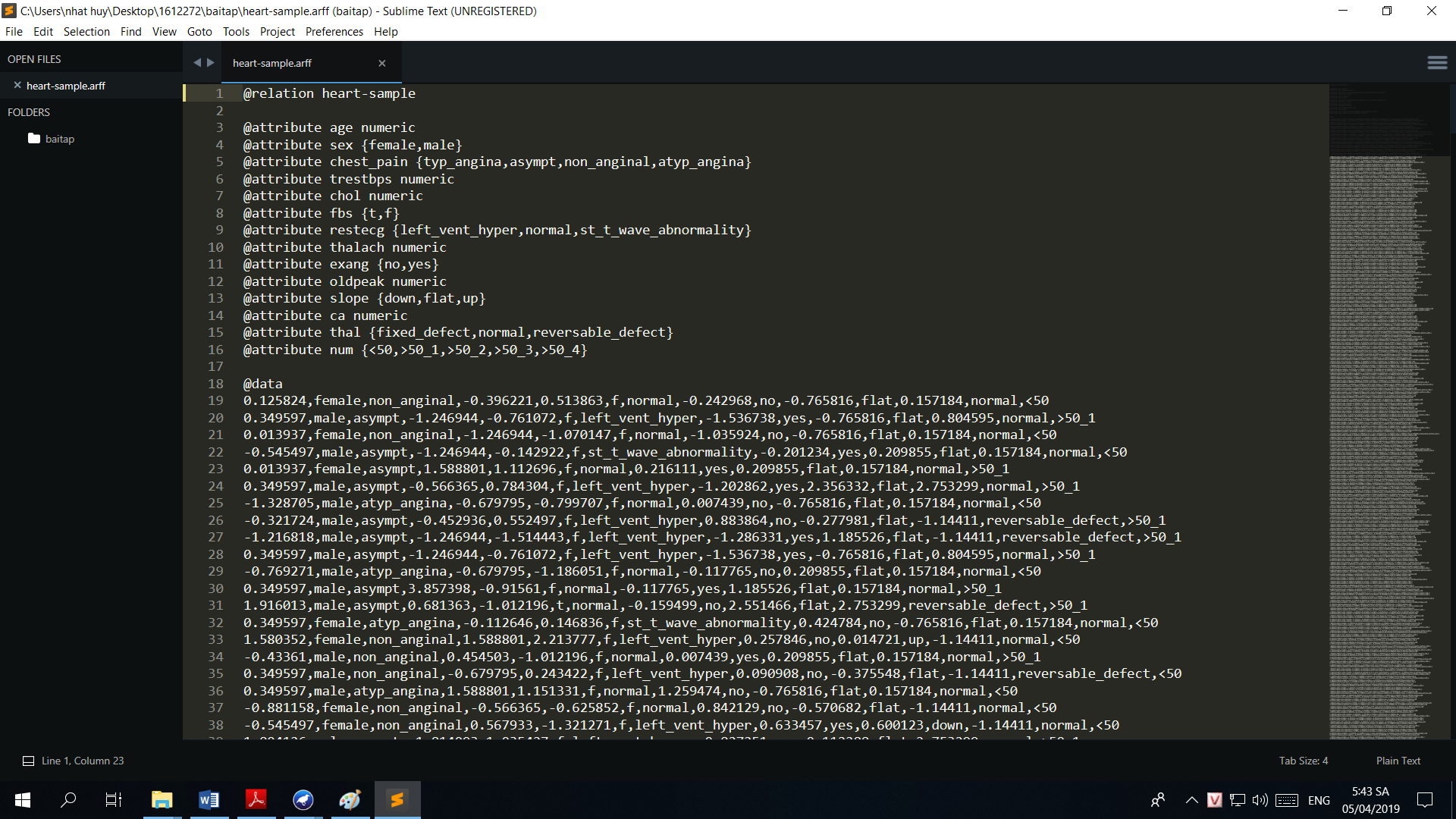
Hình 37: Ta thiết lập lấy 80% instances trong sampleSizePercent

Từ **570 instances** chúng ta còn **456 instances**



Hình 38: Kết quả sau khi lấy mẫu

Lưu lại thành file **heart-sample.arff**



Hình 39: Dữ liệu sau khi lấy mẫu