



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

Lab 01 - Preprocessing

Trần Thanh Tùng – 18120258

Trần Hữu Chí Bảo – 18120288

Môn: Khai thác dữ liệu và ứng dụng

Thành phố Hồ Chí Minh - 2020

Mục lục

Contents

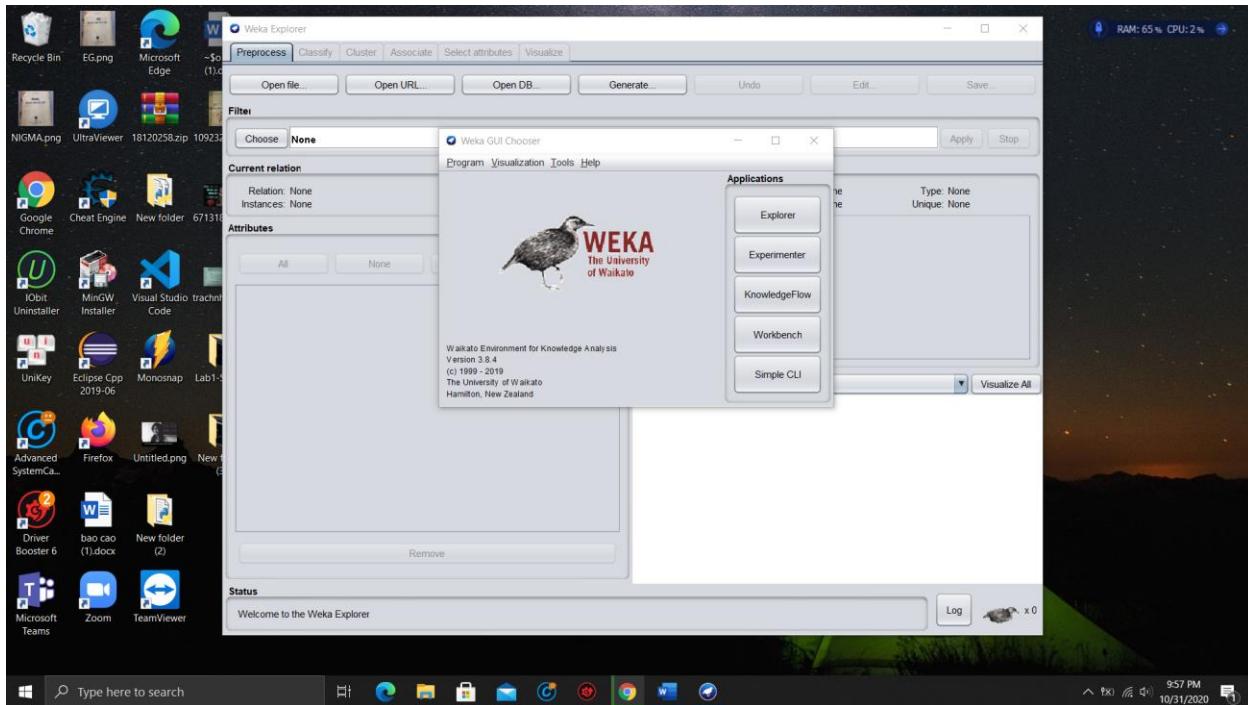
Mục lục.....	2
Đánh giá mức độ hoàn thành.....	3
1. Cài đặt Weka.....	4
1. Làm quen với Weka	6
2.1 Đọc dữ liệu vào Weka.....	6
2.2 Khám phá dữ liệu Weather	12
2.3 Khám phá tập dữ liệu Tín dụng Đức	16
3 Cài đặt tiền xử lý dữ liệu	24
3.1 Các chức năng:	24
1. Liệt kê các cột bị thiếu dữ liệu.....	24
2. Đếm số dòng bị thiếu dữ liệu.....	25
3. Điền giá trị bị thiếu bằng phương pháp mean, median (cho thuộc tính numeric) và mode (cho thuộc tính categorical). Lưu ý: khi tính mean, median hay mode các bạn bỏ qua giá trị bị thiếu.	25
4. Xóa các dòng bị thiếu dữ liệu với ngưỡng tỉ lệ thiếu cho trước (Ví dụ: xóa các dòng bị thiếu hơn 50% giá trị các thuộc tính).....	26
5. Xóa các cột bị thiếu dữ liệu với ngưỡng tỉ lệ thiếu cho trước (Ví dụ: xóa các cột bị thiếu giá trị thuộc tính ở hơn 50% số mẫu).....	27
6. Xóa các mẫu bị trùng lặp	27
7. Chuẩn hóa một thuộc tính numeric bằng phương pháp min-max và Z-score.	
28	
8. Tính giá trị biểu thức thuộc tính: ví dụ đổi với một tập dữ liệu có chứa 2 thuộc tính width và height thì biểu thức width * height sẽ trả về tập dữ liệu cũ với một thuộc tính mới có giá trị ở mỗi mẫu là tích của thuộc tính width và height trong mẫu tương ứng, với điều kiện cả 2 giá trị width và height đều không bị thiếu, trong trường hợp bị thiếu thì giá trị biểu thức coi như bị thiếu. Lưu ý: biểu thức có thể có nhiều thuộc tính và nhiều phép toán bao gồm cộng, trừ, nhân, chia.....	30

Đánh giá mức độ hoàn thành

Các mục đã hoàn thành	Mức độ hoàn thành	Người thực hiện
1.Cài đặt weka	100%	Trần Thanh Tùng 18120258
2.Làm quen với weka	100%	Trần Thanh Tùng 18120258
3. Cài đặt tiền xử lý dữ liệu	100%	Trần Hữu Chí Bảo 18120288

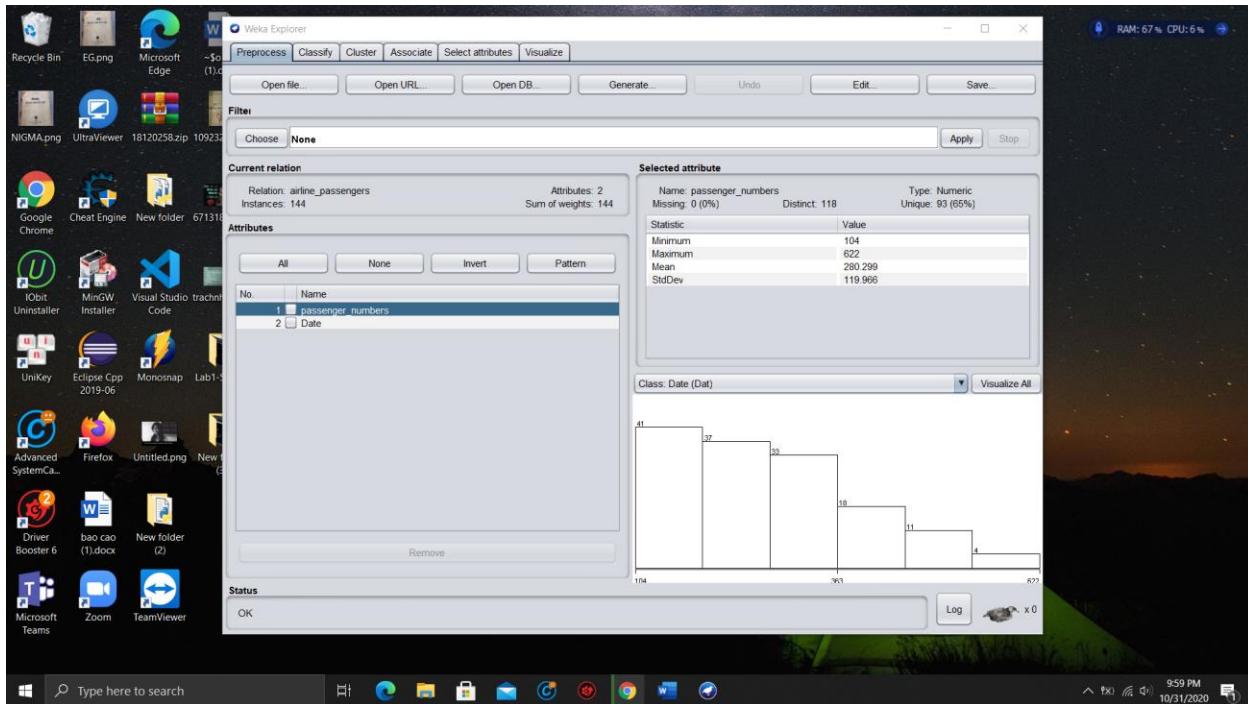
1. Cài đặt Weka

Sau khi cài đặt xong. Sinh viên chụp hình giao diện chức năng Explorer cùng màn hình desktop và báo cáo lại ảnh chụp.



Hình 1. Ảnh chụp màn hình giao diện chức năng Explorer của weka

Sinh viên tìm thư mục data trong thư mục cài đặt của Weka và mở một tập dữ liệu bất kì (có phần mở rộng là arff). Giải thích ý nghĩa các nhóm điều khiển Current relation, Attributes và Selected attribute trong tab Preprocess. Giải thích ngắn gọn ý nghĩa 5 tab trong giao diện Explorer của Weka.



Hình 2. Giao diện Weka khi mở file airline.arff

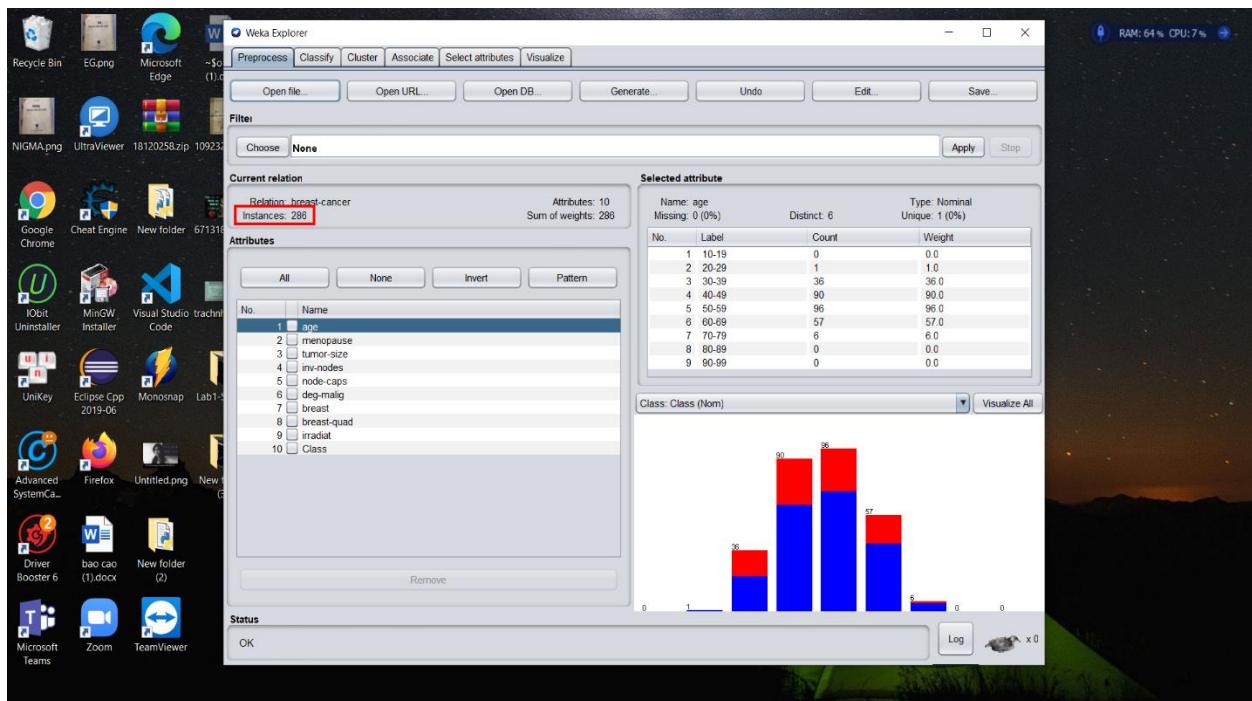
- Current relation cho ta biết các thông tin cơ bản về dữ liệu bao gồm:
 1. Relation: tên của tập giá trị dữ liệu
 2. Attributes: số tập thuộc tính của dữ liệu
 3. Instances: tổng số phiên bản của dữ liệu
 4. Sum of weight: tổng trọng số của phiên bản của dữ liệu.
- Attributes cho ta biết các thuộc tính của dữ liệu.
- Selected attribute cho ta biết các thông tin về thuộc tính đã được chọn gồm:
 1. Name : tên của thuộc tính
 2. Type: kiểu dữ liệu của thuộc tính
 3. Missing: độ thiếu của dữ liệu.
 4. Distinct: tổng số dòng dữ liệu khác nhau.
 5. Unique: tổng số dòng dữ liệu duy nhất.
- 5 tab trong giao diện Explorer của weka có ý nghĩa là:
 1. Preprocessing: Hiển thị thông số thô túc là các thông số nguyên thủy, chưa qua xử lý của dữ liệu.
 2. Classify: Phân lớp dữ liệu theo các mô hình, các chuẩn khác nhau.
 3. Cluster: Phân cụm dữ liệu theo các mô hình, các chuẩn khác nhau.
 4. Associate: Khám phá các luật kết hợp của dữ liệu.
 5. Select attribute: Quyết định các thuộc tính tương quan.
 6. Visualize: Biểu diễn trực quan dữ liệu.

1. Làm quen với Weka

2.1 Đọc dữ liệu vào Weka

1. Tập dữ liệu có bao nhiêu mẫu (instances)?

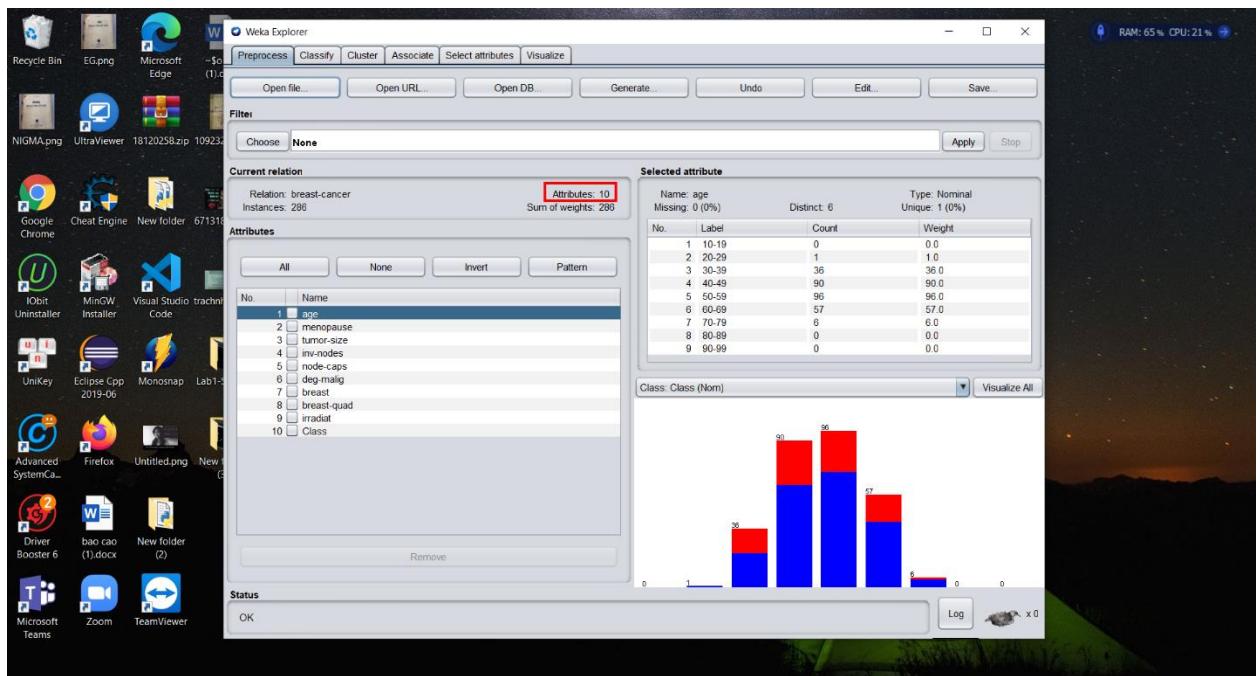
- Tập dữ liệu có 286 mẫu



Hình 3. Phần khoanh đỏ hiển thị số mẫu của dữ liệu

2. Tập dữ liệu có bao nhiêu thuộc tính (attributes)?

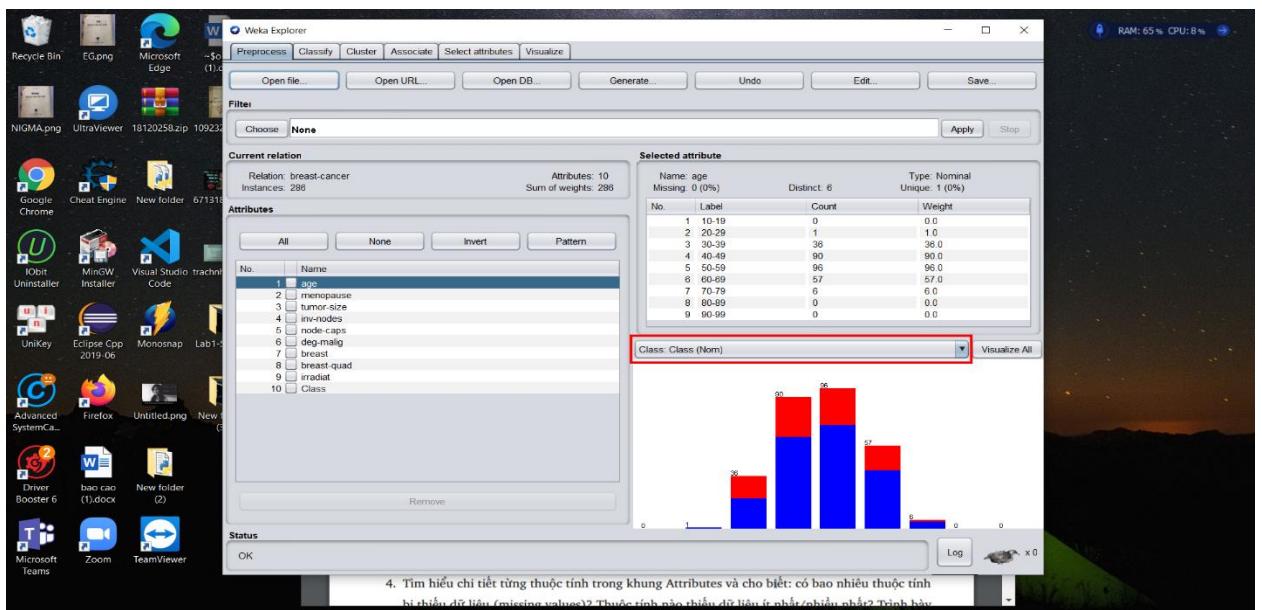
- Tập dữ liệu có 10 thuộc tính



Hình 4. Phần khoanh đỏ hiển thị số thuộc tính của dữ liệu

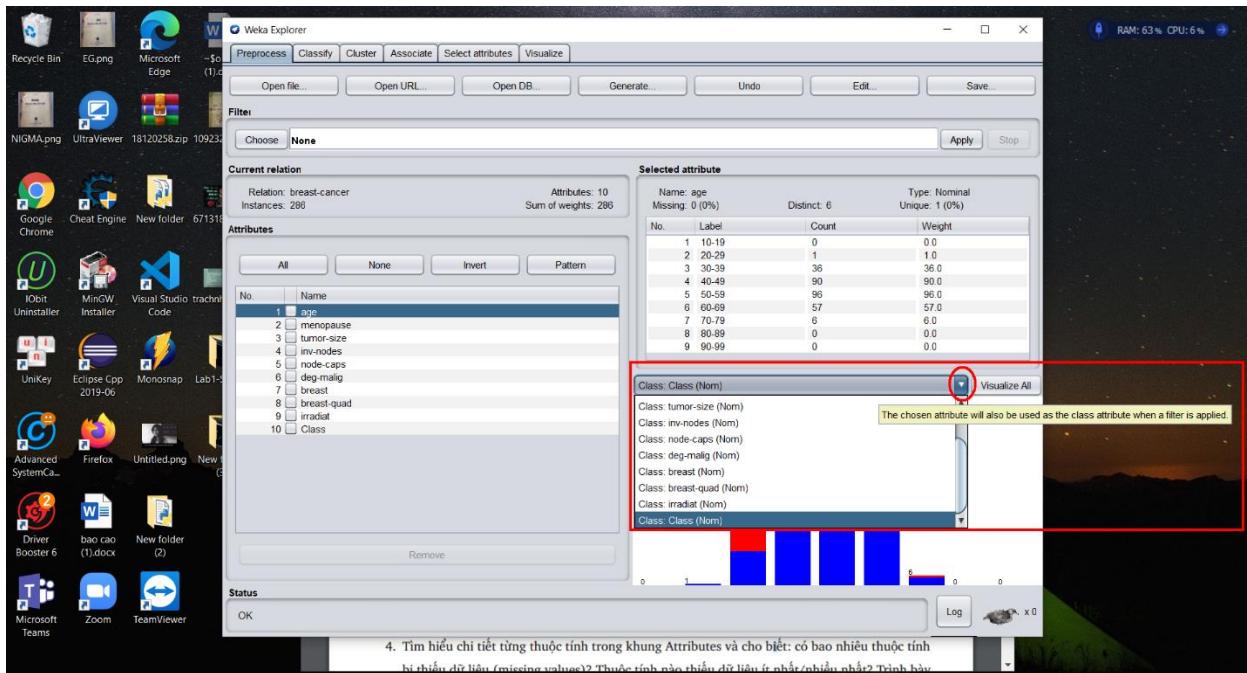
3. Thuộc tính nào được dùng làm lớp (class)? Có thể thay đổi thuộc tính dùng làm lớp hay không? Nếu có thì bằng cách nào

- Thuộc tính nào được dùng làm lớp : Class.



Hình 5. Phần khoanh đỏ hiển thị thuộc tính lớp

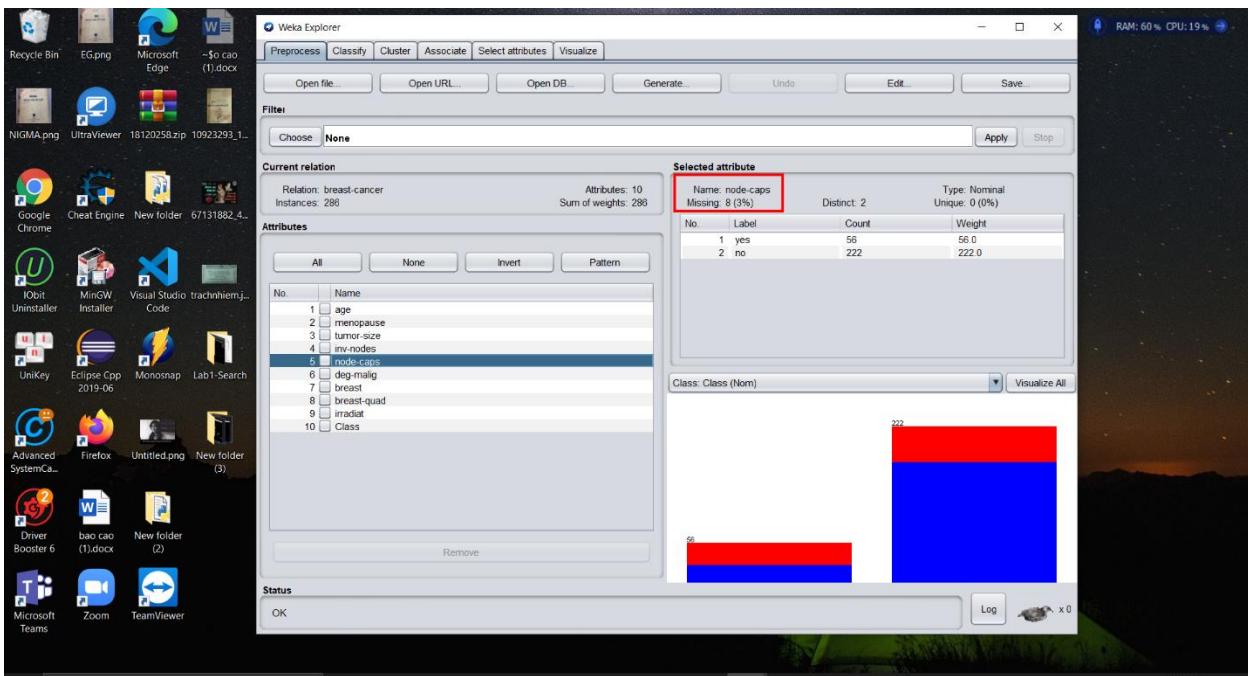
- Có thể thay đổi thuộc tính dùng làm lớp bằng cách click vào dấu mũi tên ở ô Class xong chọn thuộc tính khác.



Hình 6. *Ấn vào nút khoanh tròn màu đỏ và chọn các thuộc tính khác để làm lớp*

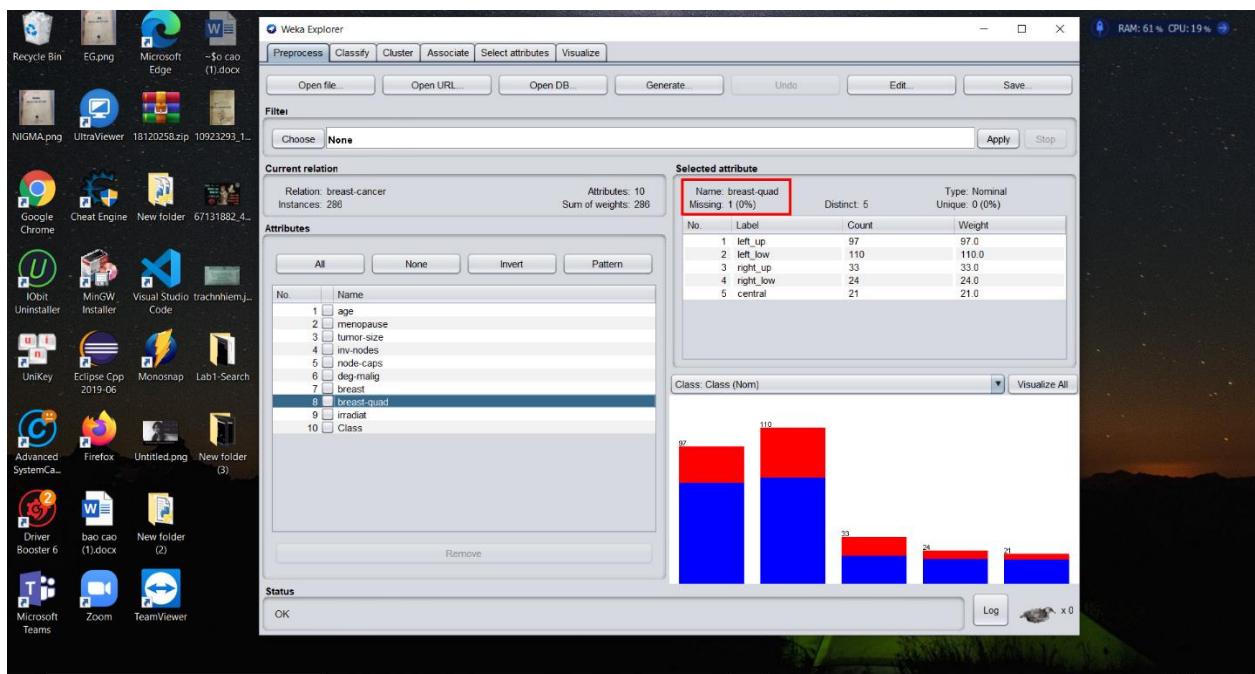
4. *Tìm hiểu chi tiết từng thuộc tính trong khung Attributes và cho biết: có bao nhiêu thuộc tính bị thiếu dữ liệu (missing values)? Thuộc tính nào thiếu dữ liệu ít nhất/nhiều nhất? Trình bày tổng quát các cách để giải quyết vấn đề missing values.*

- Có 2 thuộc tính bị thiếu dữ liệu:
- Node-caps



Hình 7. Thuộc tính Node-caps có missing rate lớp hơn 0

- Breast-quad



Hình 8. Thuộc tính Breast - quad có missing rate lớp hơn 0

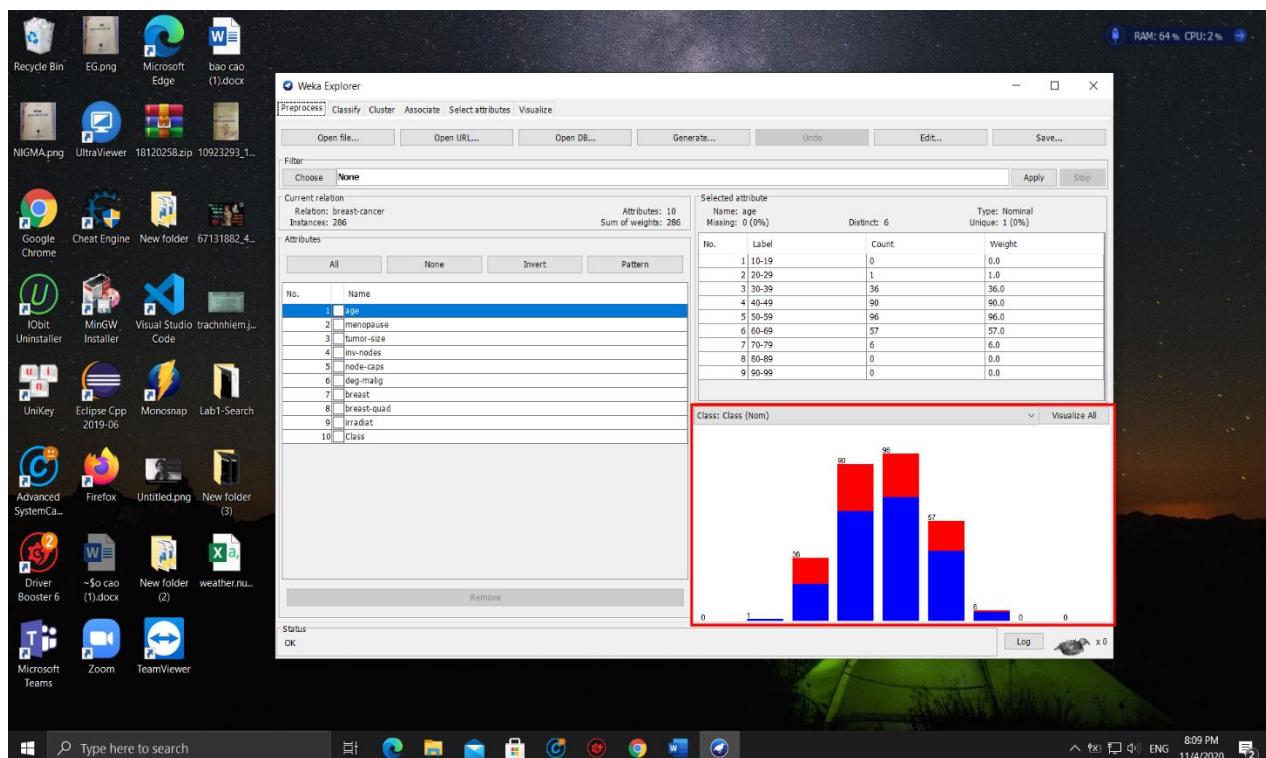
- Thuộc tính bị thiếu nhiều nhất là node-caps (8 điểm dữ liệu bị thiếu).
- Thuộc tính bị thiếu ít nhất là breast-quad (1 điểm dữ liệu bị thiếu).
- Tổng quát các cách để giải quyết vấn đề missing values:

Phương pháp	Mục đích
Loại bỏ các dòng (Ignoring the tuple)	Phương pháp này không thật sự hiệu quả, trừ khi tập dữ liệu có nhiều thuộc tính bị mất dữ liệu. Và nó thật sự không hiệu quả khi mà số điểm dữ liệu bị thiếu của các thuộc tính khác nhau đáng kể. Khi bỏ qua tập thuộc tính, ta cũng bỏ qua những dữ liệu còn lại trong tập thuộc tính, trong khi dữ liệu này có thể rất hữu ích
Thêm bằng cách thủ công (Fill in the missing value manually)	Phương pháp này rất tốn thời gian và không khả thi đối với tập dữ liệu có số lượng thiếu lớn
Dùng hằng toàn cục để điền vào các dữ liệu bị 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 $-\infty$. 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
Dùng giá trị trung tâm (mean hay median) của các thuộc tính để điền các dữ liệu bị thiếu (Use a measure of central tendency for the attribute (e.g., the mean or median) to fill in the missing value:)	Sử dụng giá trị trung tâm của thuộc tính để thay thế dữ liệu thiếu
Sử dụng giá trị mean hoặc median của thuộc tính cho tất cả các mẫu thuộc cùng một lớp như bộ đã cho (Use the attribute mean or median for all samples belonging to the same class as the given tuple.)	Bằng cách tính mean hay median các dữ liệu còn lại trong thuộc tính, ta có thể điền các giá bị thiếu bằng giá trị này để đảm bảo rằng tính chất của dữ liệu không bị thay đổi.
Dùng giá trị có khả năng nhất để điền vào dữ liệu thiếu(Use the most probable value to fill in the missing value:)	Có thể dùng các phương pháp như : hồi quy, can thiệp bằng các phương pháp theo xác suất Bayes hay tạo cây quyết định

5. Đồ thị trong cửa sổ Explorer có ý nghĩa là trực quan hóa dữ liệu của thuộc tính đang được chọn.

Đồ thị này là biểu đồ cột.

- Màu xanh tượng trưng cho các bệnh nhân không bị tái ung thư vú, màu đỏ tượng trưng cho các bệnh nhân bị tái ung thư vú.
- Đồ thị này biểu diễn cho độ tuổi các bệnh nhân bị ung thư vú, và tỷ lệ tái bệnh ở các độ tuổi.

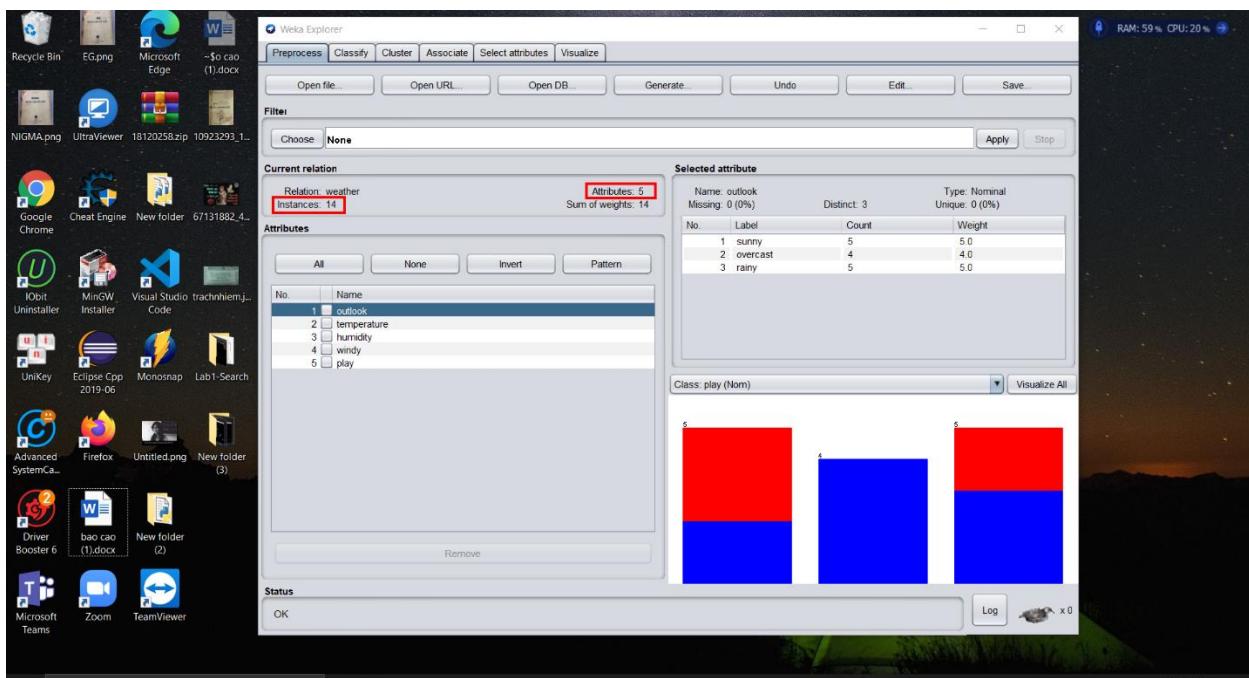


Hình 9. Đồ thị ở cửa sổ explorer

2.2 Khám phá dữ liệu Weather

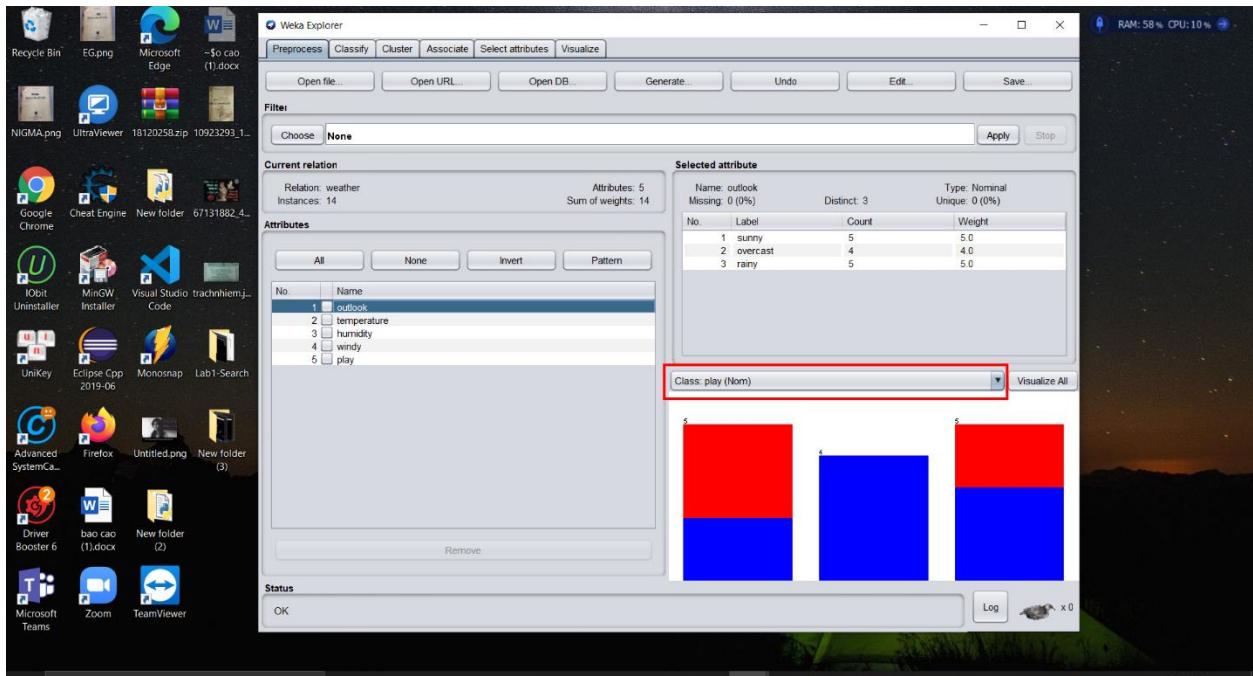
1. Tập dữ liệu có bao nhiêu thuộc tính? Bao nhiêu mẫu? Phân loại các thuộc tính theo kiểu dữ liệu (categorical/numeric). Thuộc tính nào là lớp?

 - Tập dữ liệu có 5 thuộc tính, 14 mẫu.



Hình 10. Phần khoanh đỏ hiển thị số mẫu và thuộc tính của tập dữ liệu

- Phân loại:
 - Numeric: temperature, humidity.
 - Nominal: outlook, windy, play.
 - Thuộc tính play là thuộc tính lớp.



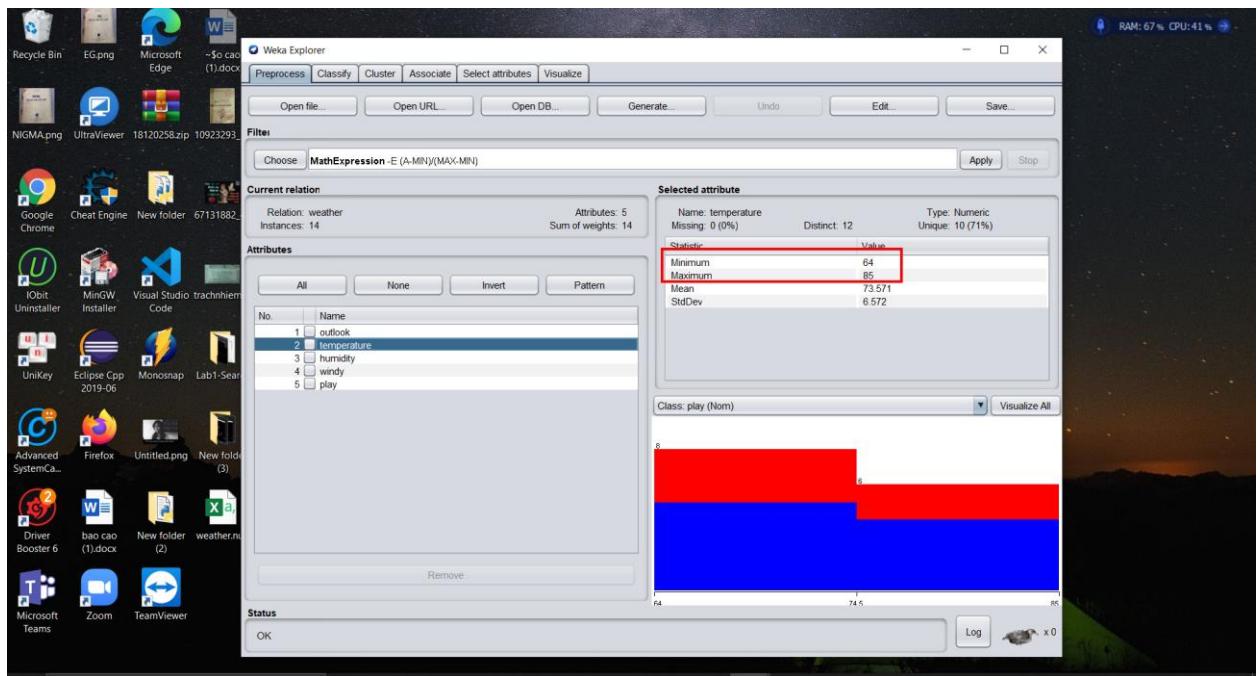
Hình 11. Thuộc tính lớp là play

2. Liệt kê five-number summary của thuộc tính temperature và humidity. Weka có cung cấp những giá trị này không?

- five-number summary của thuộc tính **temperature** và **humidity**:

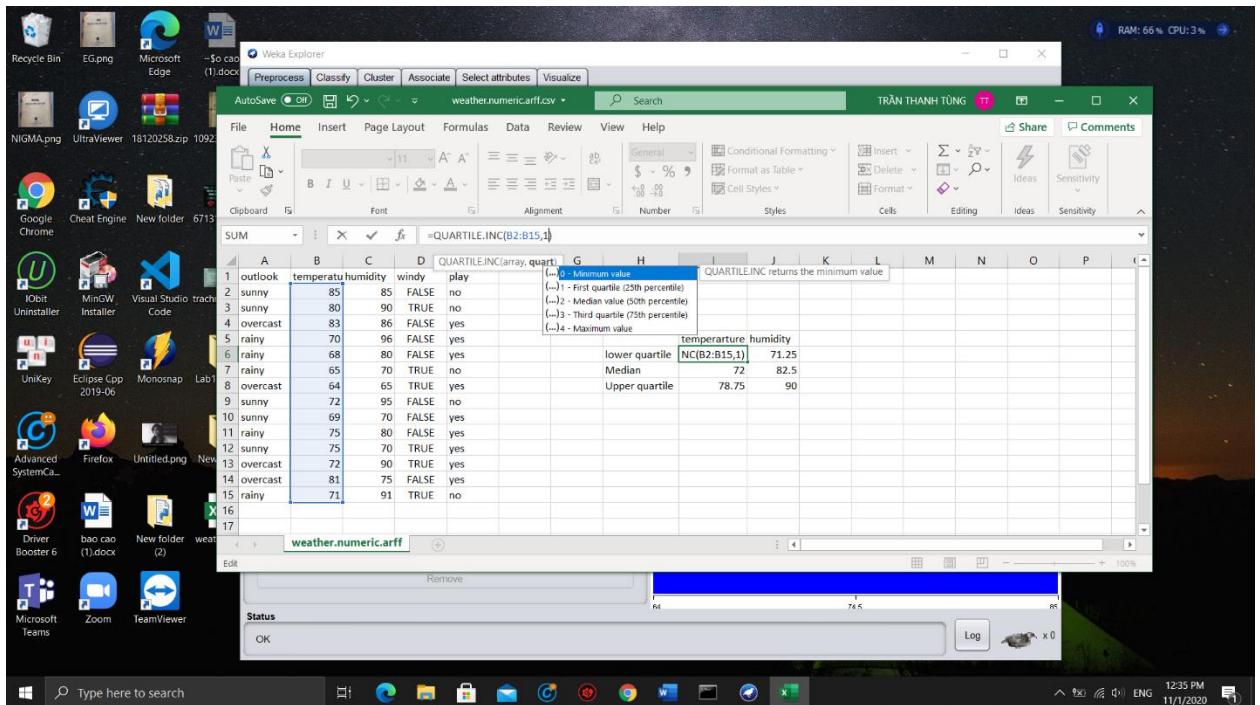
Five-number summary	Temperature	Humidity
Minimum	64	65
Lower quartile	69.25	71.25
Median	72	82.5
Upper quartile	78.75	90
Maximum	85	96

- Weka liệt kê sẵn 2 giá trị là Minimum và Maximum.



Hình 12. 2 giá trị Max và Min mà Weka liệt kê sẵn

Ở đây nhóm tính các giá trị còn lại bằng excel bằng câu lệnh quartile.inc.

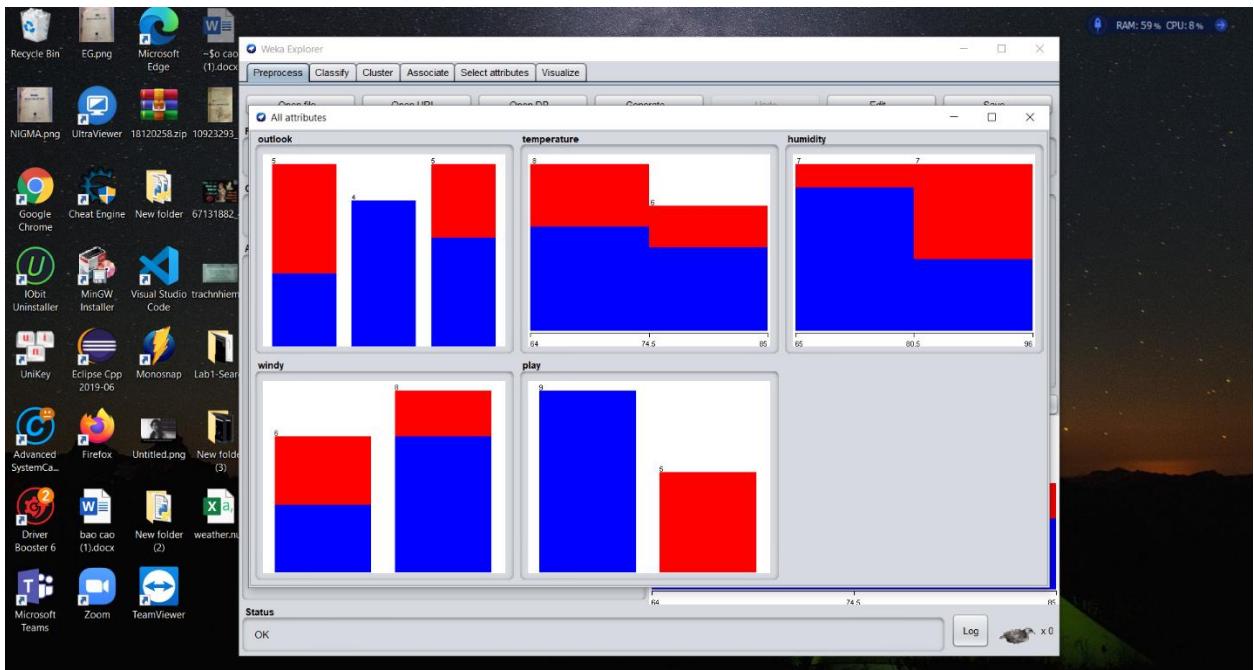


Hình 13. Sử dụng excel để tính 3 giá trị còn lại

3. *Lần lượt xem xét các thuộc tính khác của dataset dưới dạng đồ thị. Dán các ảnh chụp màn hình vào bài làm*

Các thuộc tính khác của data set dưới dạng đồ thị:

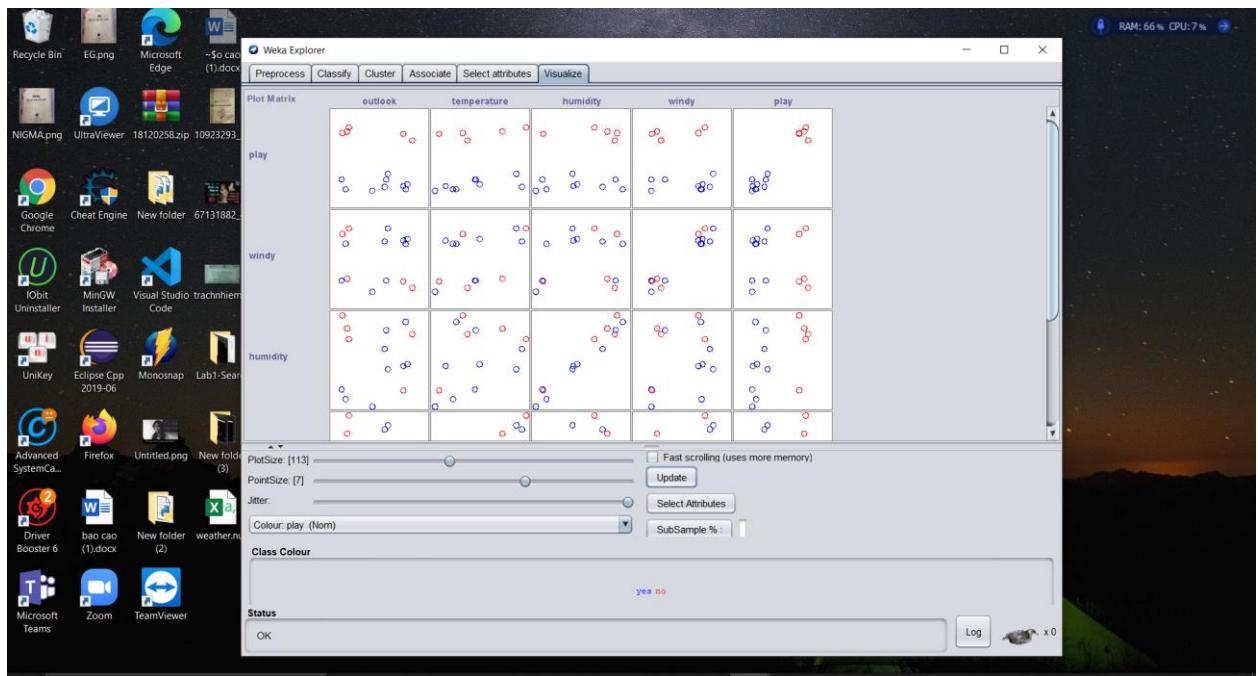
Chọn visualize all để xem tất cả các dataset dưới dạng đồ thị



Hình 14. Các đồ thị trong khi ấn vào visualize all

4. *Chuyển sang tab Visualize. Thuật ngữ sử dụng trong textbook để đặt tên cho các đồ thị ở đây là gì? Chọn jitter tối đa để thấy tổng quan hơn về phân bố dữ liệu. Theo bạn có những cặp thuộc tính khác nhau nào có vẻ như tương quan với nhau không?*

- Theo textbook, các đồ thị trong tab visualize có tên là **Scatter plot matrix**.

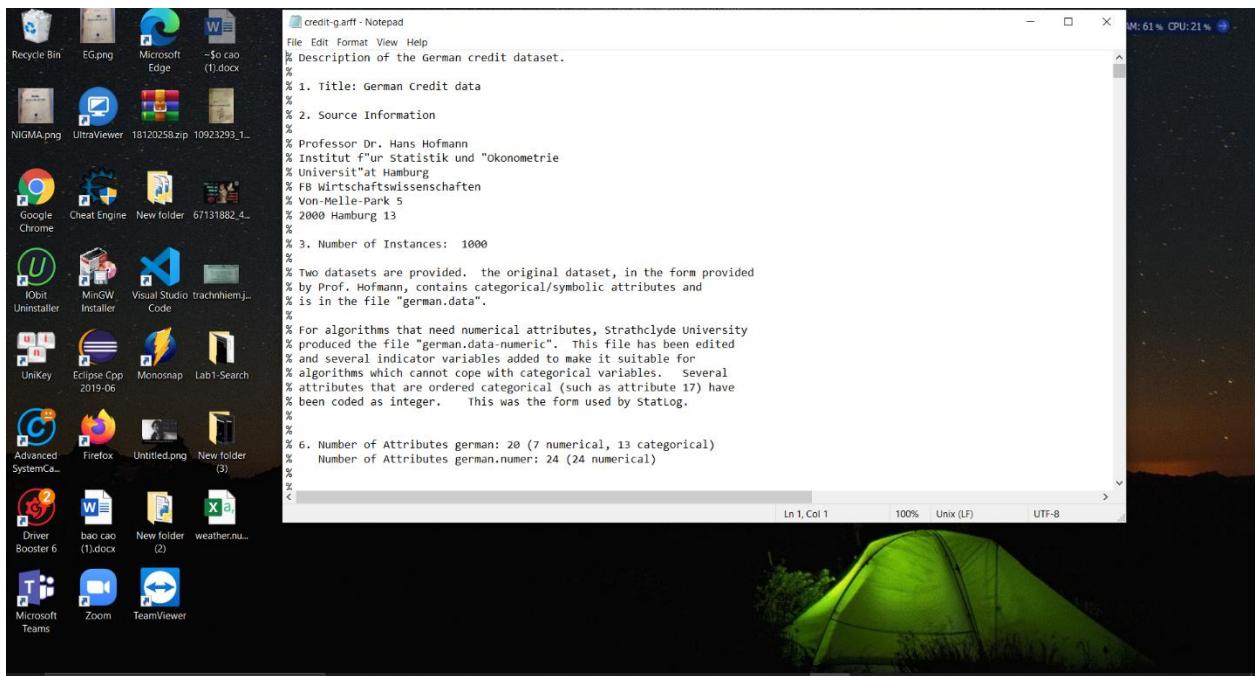


Hình 15. Các đồ Scatter plot matrix

- Các cặp thuộc tính khác nhau có vẻ tương quan với nhau là những cặp thuộc tính nằm đối xứng với nhau thông qua đường chéo phụ của ma trận.

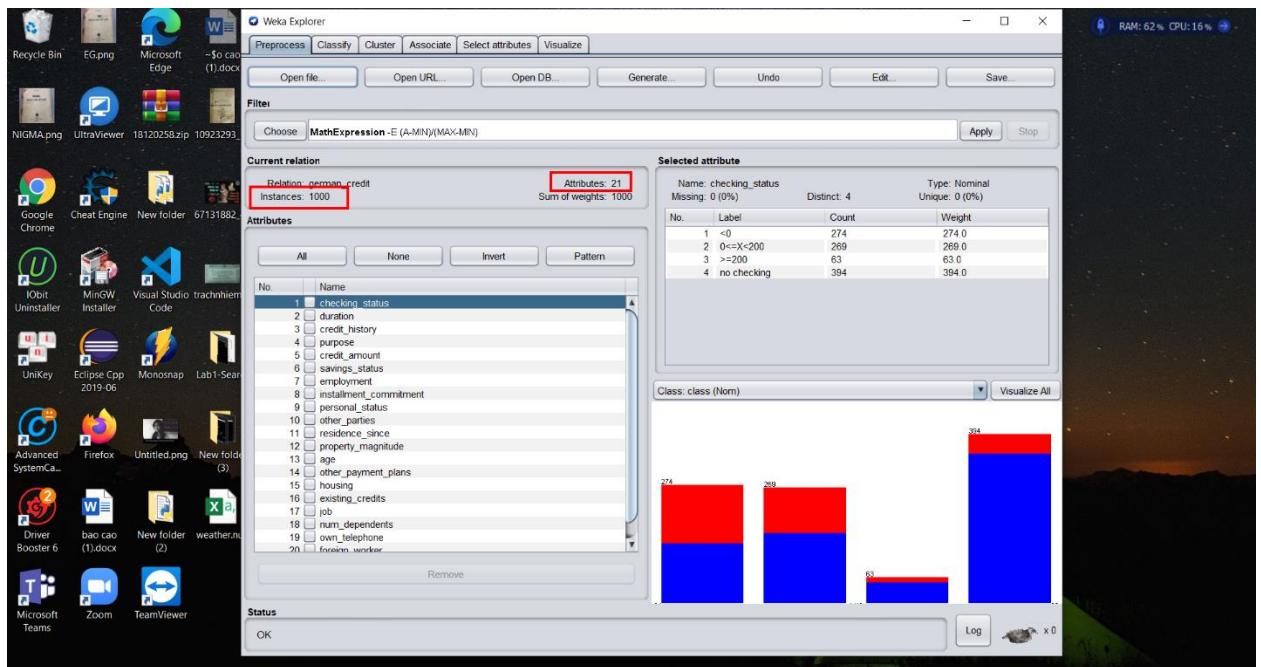
2.3 Khám phá tập dữ liệu Tín dụng Đức

1. *Nội dung của phần ghi chú (comment) trong credit-g.arff (khi mở bằng 1 text editor bất kì) nói về điều gì? Tập dữ liệu có bao nhiêu mẫu? Bao nhiêu thuộc tính? Mô tả 5 thuộc tính bất kì (phải vừa có cả thuộc tính rời rạc và thuộc tính liên tục).*
 - Khi mở file dữ liệu credit-g.arff bằng notepad, ta thấy phần ghi chú mô tả thông tin của file dữ liệu. Liệt kê tên file, nguồn, số lượng mẫu, kiểu dữ liệu các thuộc tính. Chú thích các ràng buộc cho các thuộc tính và mô tả ý nghĩa của các giá trị.



Hình 16. Mở file credit-g.arff bằng notepad

- Tập dữ liệu có 21 thuộc tính, có 1000 mẫu.



Hình 17. Phần khoanh đỏ hiển thị số mẫu và số thuộc tính của tập dữ liệu

5 thuộc tính bắt kỳ:

1. Age:

Loại dữ liệu: số

Loại thuộc tính: liên tục

Thuộc tính mô tả thông tin tuổi của khách hàng.

2. Personal status:

Loại dữ liệu: Chuỗi

Loại thuộc tính: Không liên tục

Thuộc tính mô tả giới tính và tình trạng hôn nhân của khách hàng.

3. Purpose:

Loại dữ liệu: Chuỗi

Loại thuộc tính: không liên tục.

Thuộc tính mô tả mục đích khách hàng dùng thẻ ghi nợ.

4. Own_telephone:

Loại dữ liệu: nhị phân (none/yes)

Loại thuộc tính: Không liên tục.

Thuộc tính mô tả việc khách hàng có điện thoại hay không.

5. Housing

Loại dữ liệu: Chuỗi.

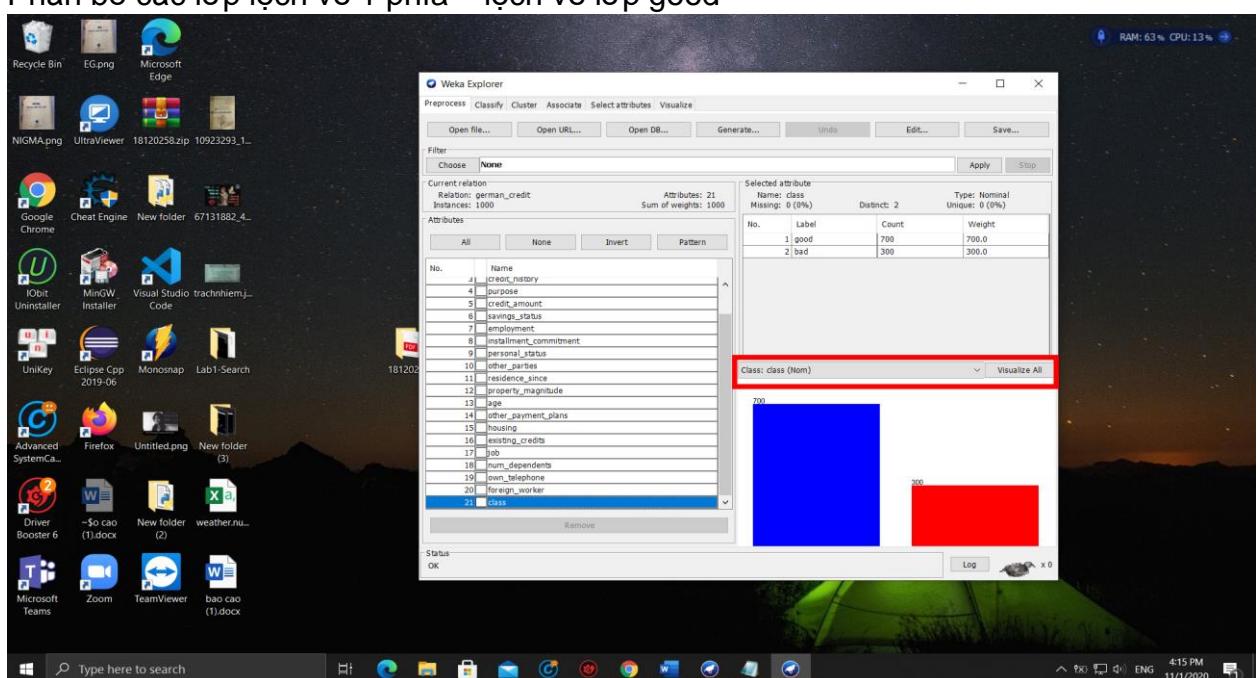
Loại thuộc tính: không liên tục.

Thuộc tính mô tả tình trạng nhà ở của khách hàng.

2. *Tên của thuộc tính lớp là gì? Đánh giá phân bố của các lớp, tức là cân bằng hay lệch về một lớp?*

- Tên của thuộc tính lớp là Class:

Phân bố các lớp lệch về 1 phía – lệch về lớp good

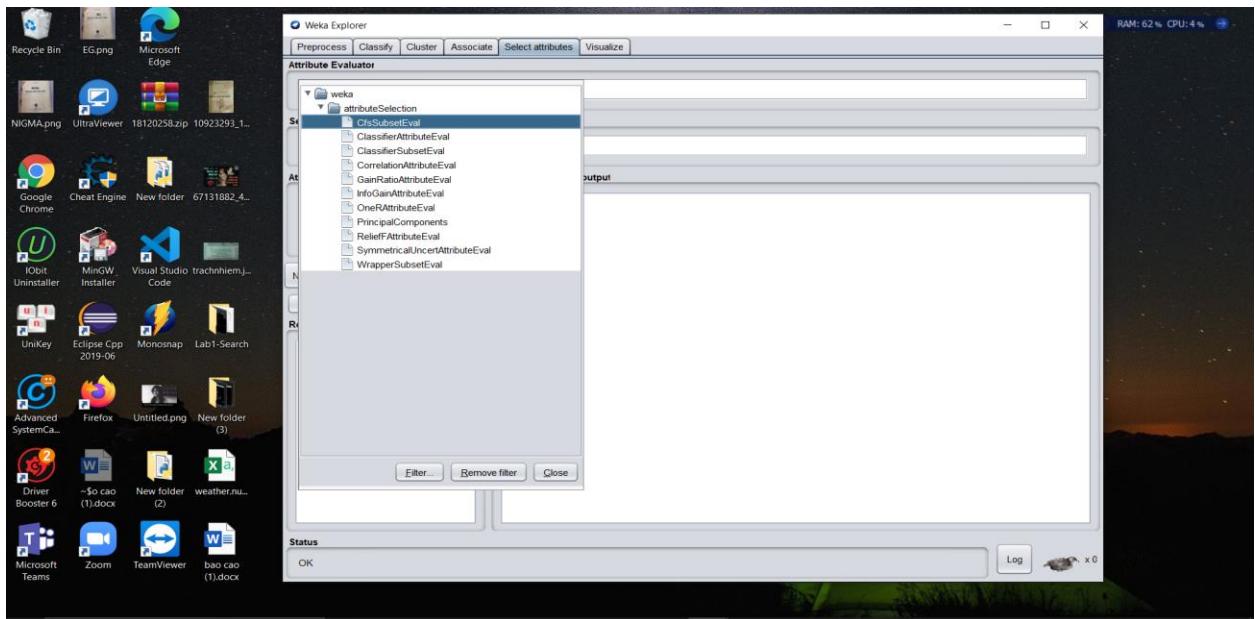


Hình 18. Đồ thị lệch về thuộc tính good

3. 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
- Các lựa chọn khác nhau của Weka để chọn lọc thuộc tính:

Phương pháp	Mục đích
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 rác hoặc giá trị của nó sẽ phụ thuộc vào phân bố của các giá trị
GainRatioAttributeEval	Đánh giá những thuộc tính bằng cách đo tỉ lệ tăng (gain ratio) của thuộc tính đó cho class
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
OneRAttributeEvalOneRAttributeEval	Sử dụng một phương pháp 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
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 riêng (eigenvector) để thỏa mãn một phương sai cho trước.
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.
SymmetricalUncertAttributeEval	Đánh giá một thuộc tính bằng cách đo symmetrical uncertainty của nó cho class
WrapperSubsetEval	Sử dụng classifier để đánh giá tập attribute và áp dụng crossvalidation để dự đoán độ chính xác của lược đồ học (learning scheme) cho từng tập

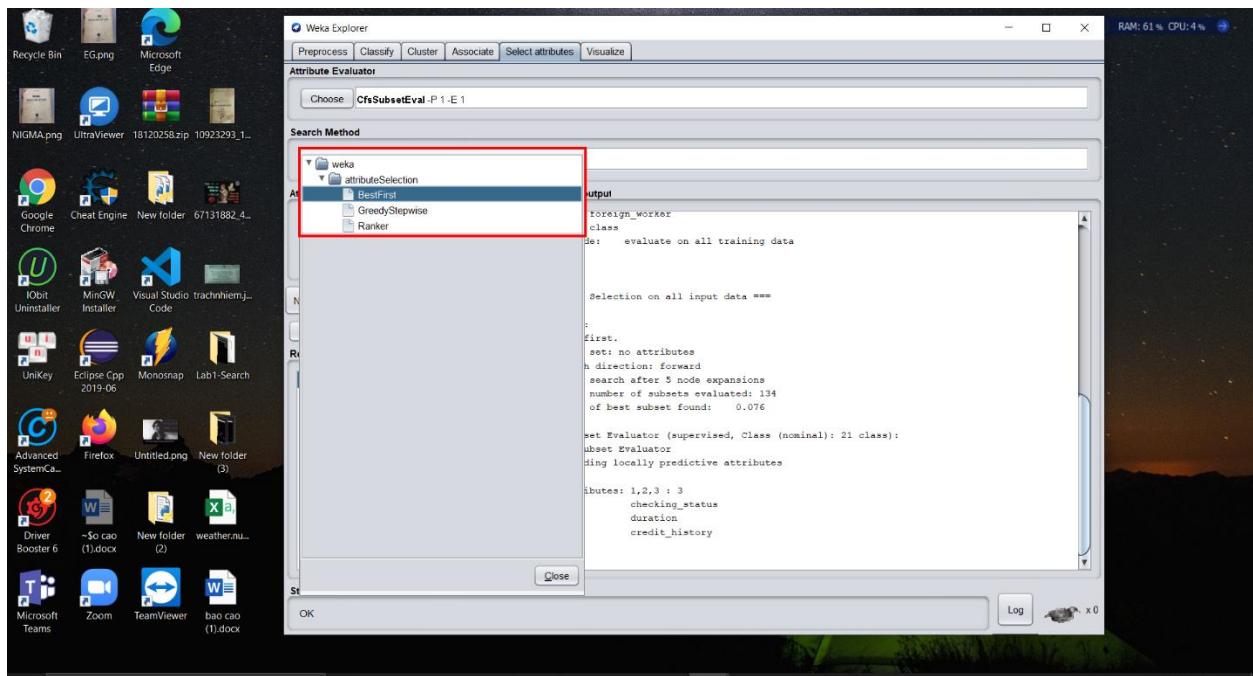
- o Các phương pháp Attribute Evaluator:



Hình 19. Các phương pháp attribute evaluator

- Các phương pháp Search method:

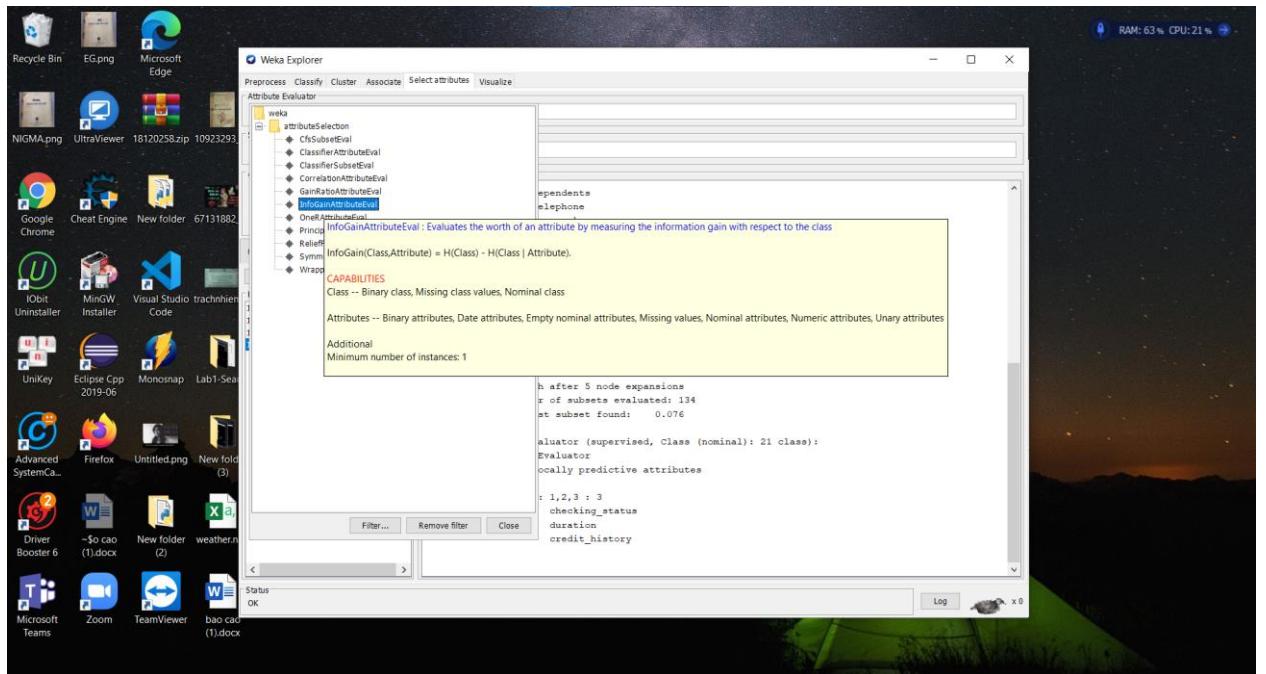
Phương pháp	Mục đích
BestFirst	Thực hiện leo đồi tham lam (greedy hill climbing) với quay lui (backtracking). Nó có thể tìm kiếm tiến (forward) từ một tập attribute rỗng, lui (backward) từ tập chứa toàn bộ attribute hoặc có thể bắt đầu từ một trạng thái cụ thể nào đó và tìm kiếm theo 2 hướng
GreedyStepwise	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
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

*Hình 20. Các phương pháp search method*

4. Cần sử dụng bộ lọc nào để chọn ra 5 thuộc tính có tương quan cao nhất với thuộc tính lớp? Mô tả các bước làm, kèm theo hình chụp từng bước và kết quả cuối cùng.

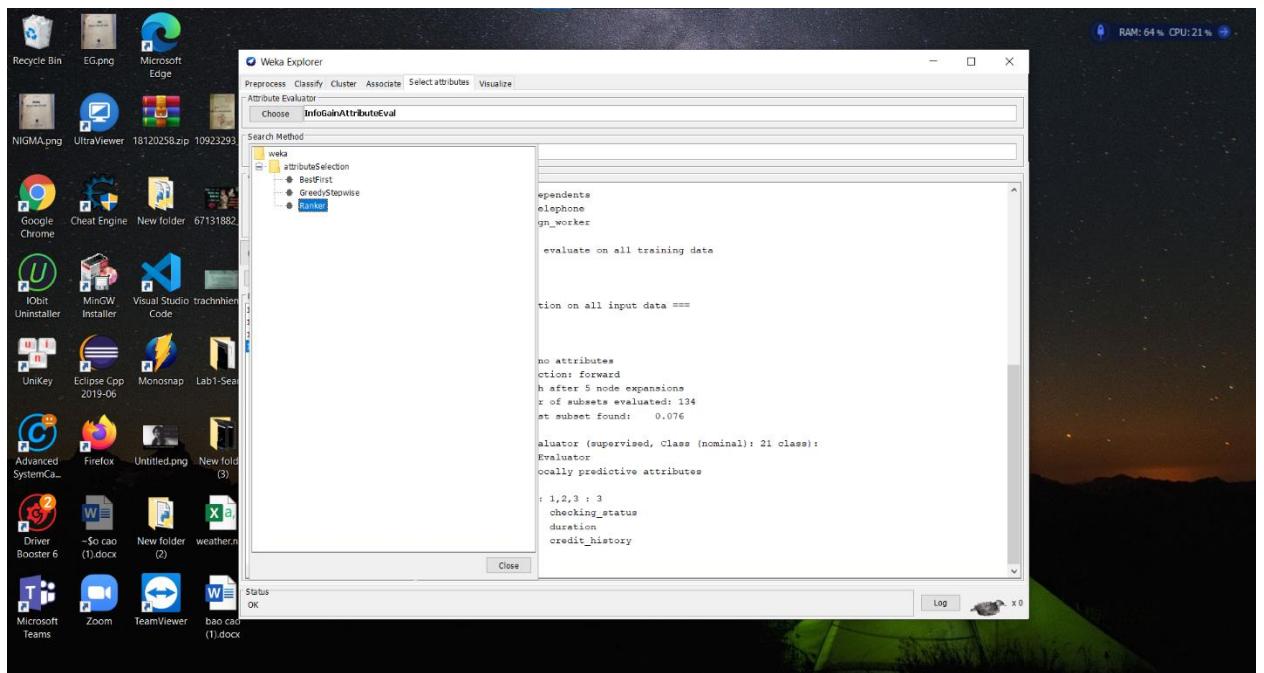
- Cần sử dụng bộ lọc InfoGainAttributeEval kết hợp với search method là Ranker để chọn ra 5 thuộc tính có tương quan cao nhất với thuộc tính lớp vì bộ lọc InfoGainAttributeEval đánh giá giá trị của một thuộc tính bằng cách đo lường thu được thông tin liên quan đến lớp kết hợp với ranker sẽ xếp hạng độ từ cao đến thấp độ tương quan của các thuộc tính với thuộc tính lớp.
- Cách làm:

1. Chọn bộ lọc InfoGainAttributeEval ở Attribute Evaluator



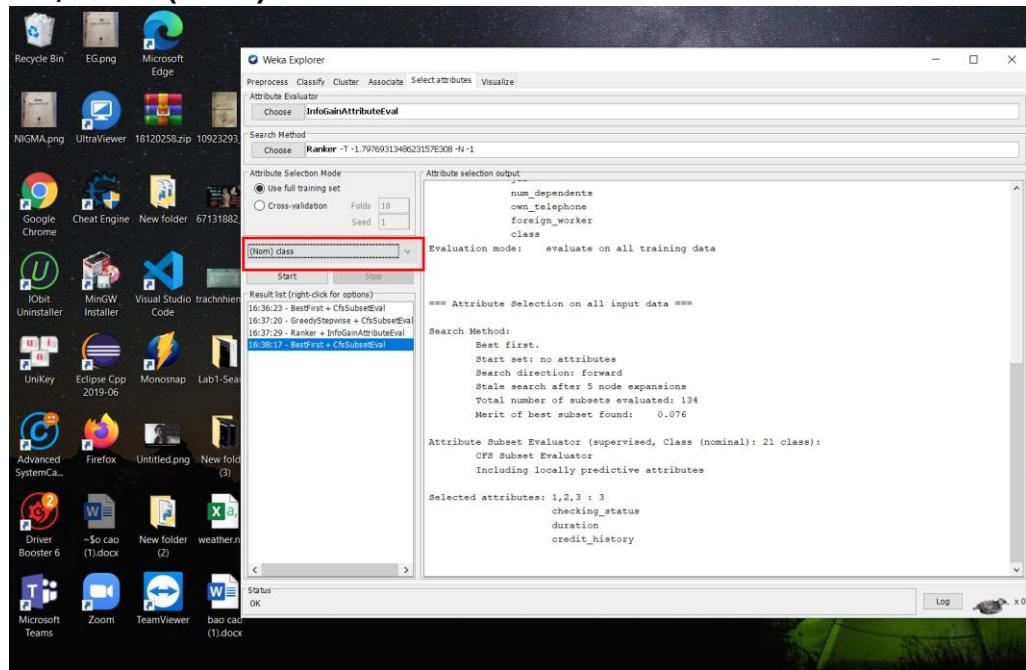
Hình 21. Chọn bộ lọc InfoGainAttribute ở Attribute evaluator

2. Chọn Ranker ở Seach method



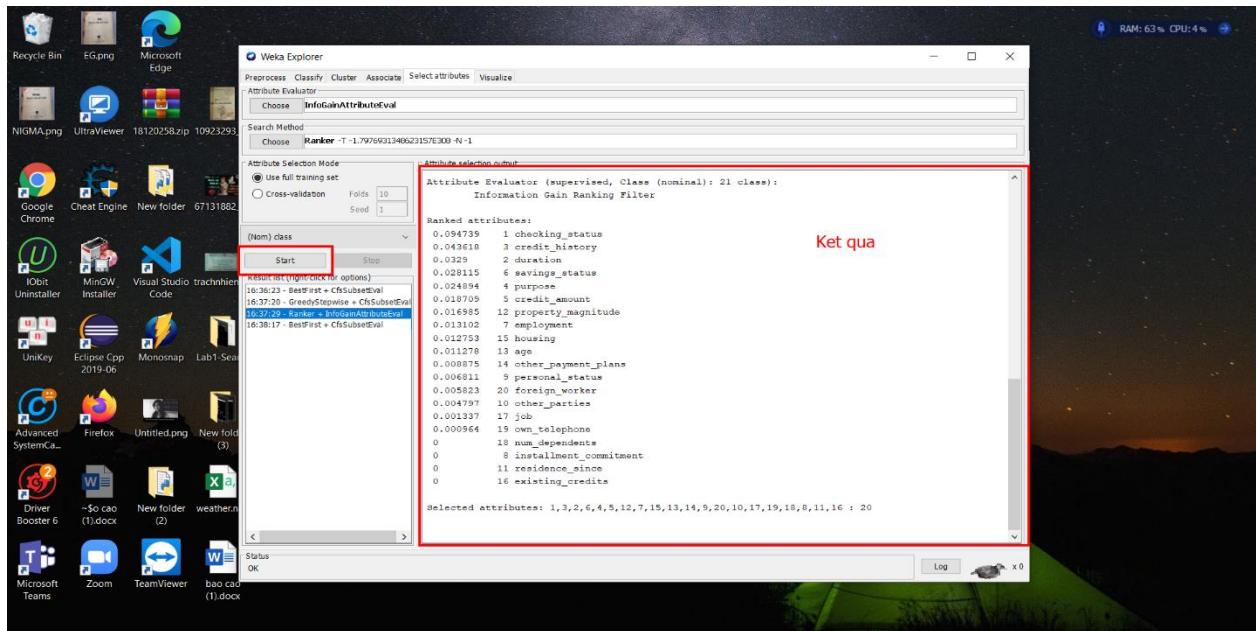
Hình 22. Chọn Ranker ở Search method

3. Chọn Nom(class)



Hình 23. Chọn Nom(class) ở vùng khoanh đỏ

4. Chọn start và nhận kết quả



Hình 24. Chọn start và nhận kết quả ở ô bên cạnh

Như vậy 5 thuộc tính có độ tương quan với lớp cao nhất là:

- | | |
|--------------------|------------------|
| 1. checking_status | 3. duration |
| 2. credit_history | 4. Saving_status |
| | 5. purpose |

3 Cài đặt tiền xử lý dữ liệu

Chương trình hoạt động theo cơ chế console và các yêu cầu người dùng được đặc tả thông qua tham số dòng lệnh.

Tham số dòng lệnh của cả chương trình được quy định như sau:

- Tham số thứ nhất là tên file thực thi, ở đây mặc định là **main.py**
- Tham số thứ hai là tên file dữ liệu cần đọc, ở đây mặc định là **house-prices.csv**
- Tham số thứ 3 là tên của các chức năng, bao gồm “cau01”, “cau02”, “cau03”, “cau04”, “cau05”, “cau06”, “cau07”.
- Đối với các hàm có xuất file, thì tham số cuối cùng là file csv có dạng **<tên file>.csv**

3.1 Các chức năng:

1. Liệt kê các cột bị thiếu dữ liệu.

Tham số dòng lệnh hoàn chỉnh của câu 1 là:

```
main.py house-prices.csv cau01
```

```

1
2     def cau01(listRow):
3         res = []
4         #tạo biến nhớ tạm để lưu kết quả
5         #đọc từng cột
6         for key in listRow[0].keys():
7             #reset biến đếm với từng cột
8             count = 0
9             #đọc từng dòng trong cột đó
10            for row in listRow:
11                if row[key] == '':
12                    res.append((key, count))
13                    count += 1
14            #chuyển biến từ list về lại dictionary ban đầu
15            res = dict(res)
16        return res

```

cau01()

2. Đếm số dòng bị thiếu dữ liệu.

Tham số dòng lệnh hoàn chỉnh của câu 2 là:

main.py house-prices.csv cau02

```
1 def cau02(listRow):
2     count=0
3     #Duyệt từng dòng
4     for row in listRow:
5         #xét từng giá trị của cột trong dòng đó
6         for key in listRow[0].keys():
7             if row[key] == '':
8                 count += 1
9                 break
10    #In ra số dòng bị thiếu dữ liệu
11    print(count)
12    return count
```

3. **Điền giá trị bị thiếu bằng phương pháp mean, median (cho thuộc tính numeric) và mode (cho thuộc tính categorical).** Lưu ý: khi tính mean, median hay mode các bạn bỏ qua giá trị bị thiếu.

Tham số thứ 4 là phương pháp điền các giá trị cho thuộc tính numeric còn thiếu: có thể mang giá trị “mean” hoặc “median”.

Tham số thứ 5 (tham số cuối cùng) là tên file csv là kết quả của tập dữ liệu sau khi điền thuộc tính còn thiếu.

Chương trình mặc định điền tất cả các thuộc tính numeric thiếu theo phương pháp người dùng chọn và các thuộc tính categorical theo phương pháp mode.

Vậy tham số dòng lệnh hoàn chỉnh của câu 3 có thể là:

main.py house-prices.csv cau03 mean Mean.csv

```

34     return attrStat
35 def calculateMean(listRow): #hàm tính mean và điền các giá trị vào cột
36     #Cập nhật các dữ liệu bị thiếu kiểu categorical
37     listRow = calculateMode(listRow)
38     mean = []
39     array_ = []
40     #xét các cột kiểu numeric mà bị thiếu dữ liệu
41     for key in attributeNumeric(listRow):
42         sum = 0
43         count = 0
44         #xét các dòng không bị thiếu dữ liệu để tính mean
45         for row in listRow:
46             if row[key] != '':
47                 count += 1
48                 sum += float(row[key])
49         average = sum / count
50         #thêm mean của cột đó vào 1 biến nhá
51         mean.append((key, average))
52     calculateMedian()

```

4. Xóa các dòng bị thiếu dữ liệu với ngưỡng tỉ lệ thiếu cho trước (Ví dụ: xóa các dòng bị thiếu hơn 50% giá trị các thuộc tính).

Tham số thứ 4 là ngưỡng tỉ lệ thiếu, có thể mang giá trị từ 0 – 100.

Tham số thứ 5 (tham số cuối cùng) là tên file csv là kết quả của tập dữ liệu sau khi xóa các dòng dữ liệu bị thiếu với ngưỡng cho trước.

Vậy tham số dòng lệnh hoàn chỉnh của câu 4 có thể là:

`main.py house-prices.csv cau04 50 deleteRow.csv`

```

1 def deleteRow(listRow, rate): #hàm xóa dòng với ngưỡng
2     temp_listRow = listRow.copy()
3     #lấy số lượng các cột
4     num_attr = len(listRow[0].keys())
5     # chuyển từ % tỉ lệ ra số thuộc tính bị thiếu
6     n = num_attr * float(rate) / 100
7     # duyệt từng dòng
8     for row in listRow:
9         count = 0
10        # duyệt từng thuộc tính trong dòng đó
11        for key in listRow[0].keys():
12            if row[key] == '':
13                #đếm số lượng thuộc tính thiếu của dòng
14                count += 1
15                #nếu số lượng thuộc tính thiếu nhiều hơn ngưỡng thì xóa dòng
16                if count > n:
17                    temp_listRow.remove(row)
18                    break
19    deleteRow()

```

5. Xóa các cột bị thiếu dữ liệu với ngưỡng tỉ lệ thiếu cho trước (Ví dụ: xóa các cột bị thiếu giá trị thuộc tính ở hơn 50% số mẫu).

Tham số thứ 4 là ngưỡng tỉ lệ thiếu, có thể mang giá trị từ 0 – 100.

Tham số thứ 5 (tham số cuối cùng) là tên file csv là kết quả của tập dữ liệu sau khi xóa các cột dữ liệu bị thiếu với ngưỡng cho trước.

Vậy tham số dòng lệnh hoàn chỉnh của câu 5 có thể là:

```
main.py house-prices.csv cau05 50 deleteColumn.csv
```

```

1  def deleteColumn(listRow, rate):
2      #lấy số lượng các dòng
3      num_attr = len(listRow)
4      # chuyển từ % ngưỡng tỉ lệ ra số dữ liệu cụ thể của cột
5      n = num_attr * float(rate) / 100
6      #tạo biến mới lưu lại tập dữ liệu
7      temp_listRow = []
8      #tạo ra 1 bản copy của dữ liệu ban đầu
9      for i in range(len(listRow)):
10         b = listRow[i].copy()
11         temp_listRow.append(b)
12         #đọc từng cột
13         for key in listRow[0].keys():
14             t = 0
15             #với mỗi cột thì reset biến đếm
16             count = 0
17             #đọc từng dòng trong cột
18             for row in listRow:
                deleteColumn() > for key in listRow[0].keys() > for row in listRow > if count > n > for index in range(len(listRow))

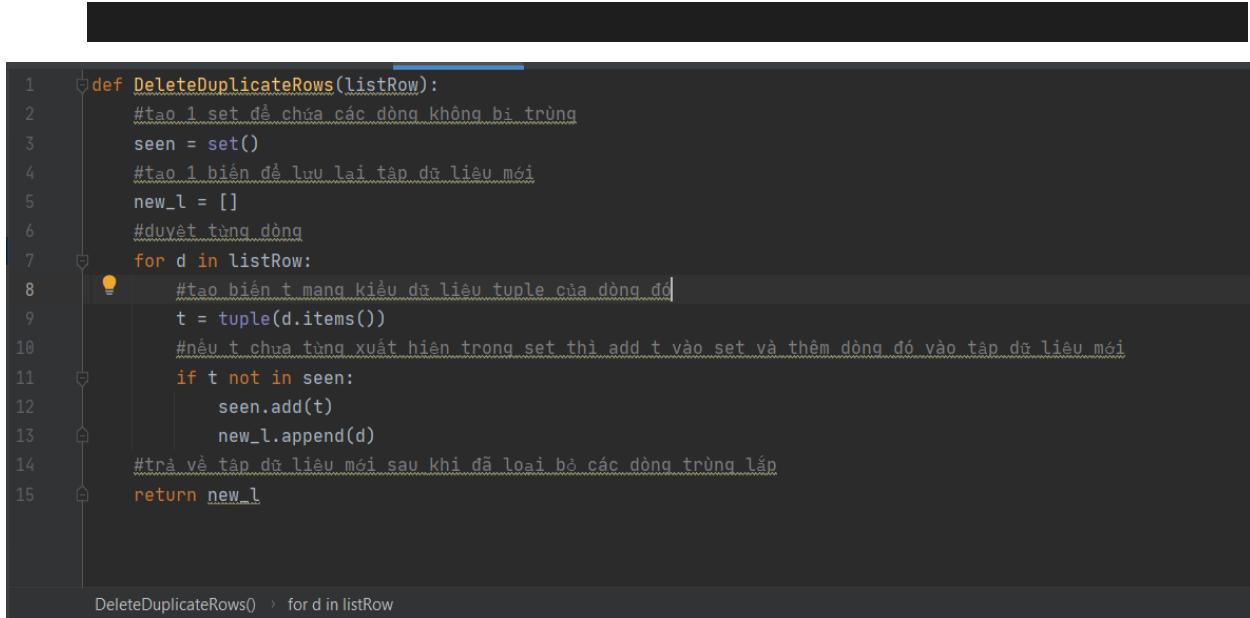
```

6. Xóa các mẫu bị trùng lặp

Tham số thứ 4 (tham số cuối cùng) là tên file csv là kết quả của tập dữ liệu sau khi xóa các dòng dữ liệu bị trùng.

Vậy tham số dòng lệnh hoàn chỉnh của câu 6 có thể là:

```
main.py house-prices.csv cau06 deleteDuplicateRows.csv
```



```
1 def DeleteDuplicateRows(listRow):
2     #tạo 1 set để chứa các dòng không bị trùng
3     seen = set()
4     #tạo 1 biến để lưu lại tập dữ liệu mới
5     new_l = []
6     #duyệt từng dòng
7     for d in listRow:
8         #tạo biến t mang kiểu dữ liệu tuple của dòng đó
9         t = tuple(d.items())
10        #nếu t chưa từng xuất hiện trong set thì add t vào set và thêm dòng đó vào tập dữ liệu mới
11        if t not in seen:
12            seen.add(t)
13            new_l.append(d)
14        #trả về tập dữ liệu mới sau khi đã loại bỏ các dòng trùng lặp
15    return new_l
```

7. Chuẩn hóa một thuộc tính numeric bằng phương pháp min-max và Z-score.

Tham số thứ 4 phương pháp sử dụng để chuyển đổi thuộc tính.

Tham số thứ 5 phương pháp sử dụng để lưu kết quả của phương pháp chuyển đổi.

Vậy tham số dòng lệnh hoàn chỉnh của câu 7 có thể là:

```
python main.py house-prices.csv cau07 Min-Max Id-OverallQual Min-Max.csv
python main.py house-prices.csv cau07 Z-Score Id-MSSubClass-OverallQual Z-
Score.csv
```

```

65     def setAttributeByZ_ScoreMethod(listRow, chuoit):
66         temp = []
67         check1 = 0
68         check = 0
69         for attr in chuoit.split("-"):
70             if attr not in allNumeric(listRow): #kiem tra attr co phai thuoc tinh so hay khong
71                 check1 += 1
72                 if check1 <= len(chuoit.split('-')):
73                     print(attr + " Không phải là một thuộc tính số hoặc không có trong danh sách các thuộc tính.")
74                     if check1 == len(chuoit.split('-')):
75                         return 0
76             else:
77                 temp_listRow = []
78                 name = attr + " Z-Score" # ten thuoc tinh chuyen doi cua thuoc tinh dang xet
79                 for i in range(len(listRow)):
80                     b = listRow[i].copy()
81                     temp_listRow.append(b)
82                     if attr in Cau03.attributeNumeric(listRow):
83                         xich_ma = 0
84                         n = len(listRow) - Cau01.cau01(listRow)[attr]
85                         #tinh mean cua thuoc tinh dang xet
86                         count = 0
87                         tong = 0
88                         for row in listRow:
89                             if row[attr] != '':
90                                 count += 1
91                                 tong += float(row[attr])
92                         x = tong / count
93                         for row in temp_listRow:
94                             if row[attr] != '':
95                                 xich_ma += 1/(n-1)*pow((float(row[attr]) - x), 2) #cong thu tinh do lech chuan
setAttributeByMin_MaxMethod() > for attr in chuoit.split("-") > else > if check == 0 > for row in temp_listRow > if row[attr] == ""

```

```

18     def setAttributeByMin_MaxMethod(listRow, chuoit):
19         temp = []
20         check = 0
21         check1 = 0
22         for attr in chuoit.split("-"): # truy xuat tung thuoc tinh trong 1 chuoi
23             if attr not in allNumeric(listRow):
24                 check1 += 1
25                 if check1 <= len(chuoit.split('-')):
26                     print(attr + " Không phải là một thuộc tính số hoặc không có trong danh sách các thuộc tính.")
27                     if check1 == len(chuoit.split('-')):
28                         return 0
29             else:
30                 list1 = []
31                 #sao chep temp lisRow tu listRow
32                 temp_listRow = []
33                 for row in listRow:
34                     if row[attr] != '':
35                         list1.append(float(row[attr]))
36                 max1 = max(list1)
37                 min1 = min(list1)
38                 t = max1 - min1
39                 name = attr + " Min-Max" #cot chuyen hoa tuong ung voi thuoc tinh do
40                 for i in range(len(listRow)):
41                     b = listRow[i].copy()
42                     temp_listRow.append(b)
43                     if check == 0:
44                         for row in temp_listRow:
45                             if row[attr] == '':
46                                 temp.append({attr: row[attr], name: ''})
47                             else:
48                                 value = (float(row[attr]) - min1) / t #gia tri chuyen doi (x-min)/(max-min)
49                                 temp.append({attr: row[attr], name: value})
setAttributeByMin_MaxMethod() > for attr in chuoit.split("-") > else > if check == 0 > for row in temp_listRow > if row[attr] == ""

```

8. Tính giá trị biểu thức thuộc tính: ví dụ đổi với một tập dữ liệu có chứa 2 thuộc tính width và height thì biểu thức width * height sẽ trả về tập dữ liệu cũ với một thuộc tính mới có giá trị ở mỗi mẫu là tích của thuộc tính width và height trong mẫu tương ứng, với điều kiện cả 2 giá trị width và height đều không bị thiếu, trong trường hợp bị thiếu thì giá trị biểu thức coi như bị thiếu. Lưu ý: biểu thức có thể có nhiều thuộc tính và nhiều phép toán bao gồm cộng, trừ, nhân, chia.

Tham số thứ 4 phương pháp sử dụng để chuyển đổi thuộc tính.

Tham số thứ 5 phương pháp sử dụng để lưu kết quả của phương pháp chuyển đổi.

Vậy tham số dòng lệnh hoàn chỉnh của câu 8 có thể là:

```
python main.py house-prices.csv cau08 Id+LotFrontage*OverallQual-
YearRemodAdd/OverallCond AddColumn.csv
```

```

27 def addColumn(listRow, chuoi):
28     #tao mang temp_listRow sao chep du lieu tu listRow
29     temp_listRow = []
30     for i in range(len(listRow)):
31         b = listRow[i].copy()
32         temp_listRow.append(b)
33         x = XuLyChuoi(listRow, chuoi)
34         temp = []
35         count = 0
36         if x == 0:
37             return 0
38         else:
39             for row in temp_listRow:
40                 sum = []
41                 check = 0
42                 #ham anh xa cac thuộc tính trong 1 dong lieu thanh cac gia tri tương ứng
43                 for i in range(len(x)):
44                     if x[i] == '+' or x[i] == '-' or x[i] == '*' or x[i] == '/':
45                         sum.append(x[i])
46                     else:
47                         if row[x[i]] == '':
48                             check = 1
49                             temp_listRow[count][chuoi] = ''
50                             break
51                         else:
52                             sum.append(row[x[i]])
53                 if check == 0:
54                     #ham tinh toan gia tri cua bieu thuc so hoc
55                     t = eval(''.join(sum))
56                     temp_listRow[count][chuoi] = t #them thuộc tính vào trong listRow
57                     count += 1
58     return temp_listRow

```