TRƯỜNG ĐẠI HỌC THỦY LỢI

KHOA CÔNG NGHỆ THÔNG TIN

KHAI PHÁ DỮ LIỆU

Đề tài:

Khai phá dữ liệu Mức độ đánh giá chất lượng ô tô bằng phương pháp phân lớp

Giảng viên: Trần Mạnh Tuấn

Nhóm sinh viên thực hiện: Nhóm 03

1. Lưu Việt Anh – 2051063802
2. Nguyễn Vũ Phương Anh – 2051063828
3. Bùi Tiến Dũng – 2051063830
4. Nguyễn Thế Hiếu – 2051060509

HÀ NỘI, 2023

MỤC LỤC

LỜI CẢM ƠN

Ngày nay, việc ứng dụng công nghệ thông tin đã trở nên phổ biến trong hầu hết mọi cơ quan, doanh nghiệp, trường học đặc biệt là việc áp dụng các giải pháp tin học trong công tác quản lý. Trong ít năm trở lại đây, với tốc độ phát triển như vũ bão, CNTT đang dần làm cho cuộc sống của con người trở nên thú vị và đơn giản hơn. Để bắt kịp với nhịp độ phát triển của xã hội, những kiến thức học được trên giảng đường là vô cùng quan trọng đối với mỗi sinh viên chúng em. Vì vậy chúng em chọn đề tài ***"Khai phá dữ liệu Mức độ đánh giá chất lượng ô tô bằng phương pháp phân lớp"*** để làm báo cáo kết thúc môn học của mình.

Chúng em chân thành xin gửi lời cảm ơn đặc biệt đến thầy giáo Trần Mạnh Tuấn – người đã tận tình giảng dạy môn Khai phá dữ liệu cho chúng em trong từng buổi học. thầy đã giúp chúng em trang bị kiến thức môn học và hơn cả là động lực tiếp tục trên con đường chinh phục công nghệ.

Bên cạnh những kết quả mà chúng em đạt được thì sẽ không khó tránh khỏi những thiếu sót trong quá trình làm đề tài vì thời gian không cho phép và chưa có kinh nghiệm thực tế. Chính vì vậy chúng em rất mong được sự cảm thông, chỉ bảo góp ý của thầy cô. Những lời nhận xét, góp ý của thầy cô chính là một bài học, kiến thức cho chúng em trên con đường sau này.

Chúng em xin chân thành cảm ơn!

I. Mô tả bài toán

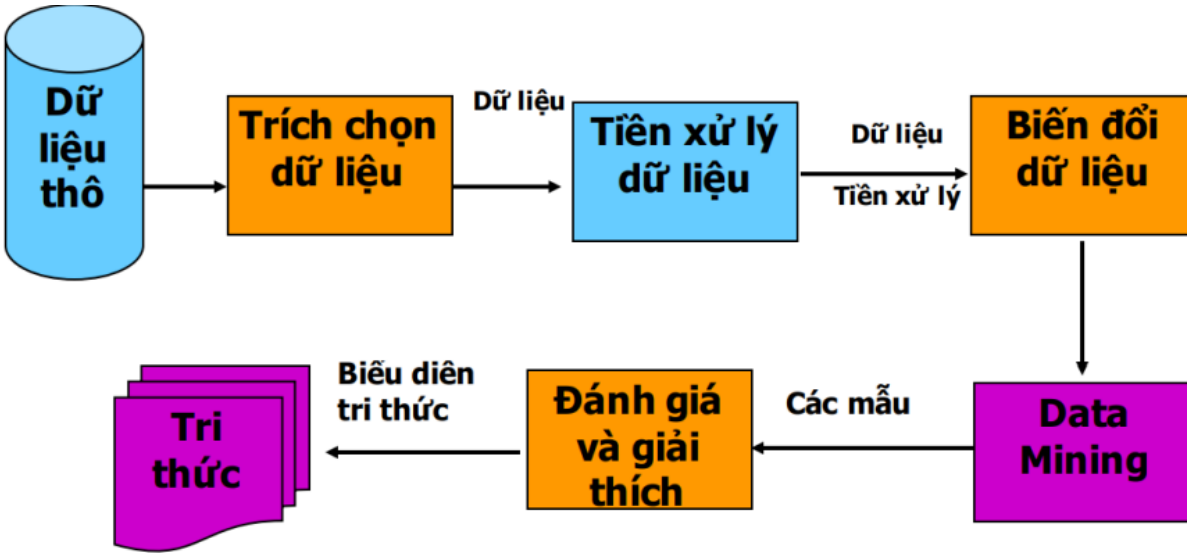
1. Lý do chọn đề tài

Những năm gần đây nhu cầu mua sắm xe ô tô của người Việt trở nên rầm rộ hơn bao giờ hết, một phần vì do cơ chế mở cửa của nhà nước, một phần vì nhu cầu phục vụ của phương tiện này ngày càng lớn, có người mua xe để phục vụ đi lại, người mua xe phục vụ ngành vận tải, làm ăn, có người thì lại mua ô tô chỉ để thể hiện đẳng cấp... . Bên cạnh đó, thực trạng hiện nay các vụ tai nạn ô tô xảy ra mỗi ngày với nhiều lý do khác nhau nhưng phần trăm do xe chưa đảm bảo an toàn là rất cao. Vì vậy, nhóm em chọn đề tài ***"Khai phá dữ liệu Mức độ đánh giá chất lượng ô tô con bằng phương pháp phân lớp"*** để thực hiện đánh giá chất lượng chiếc xe.

Tổng quan bài toán

Dataset bao gồm các mô tả về các thuộc tính tương ứng với mức độ đánh giá chất lượng của ô tô con. Áp dụng các thuật toán để xác định xem chiếc ô tô đó có chất lượng: rất tốt (vgood), tốt (good), chấp nhận (acc), không chấp nhận (unacc)

2. Quy trình thực hiện



- Quy trình thực hiện khai phá bao gồm 6 bước:
  - Bước 1: Tạo tập tin dữ liệu đầu vào
  - Bước 2: Tiền xử lý, làm sạch tập dữ liệu
  - Bước 3: Chọn tác vụ khai phá dữ liệu (phân lớp)
  - Bước 4: Khai phá dữ liệu: tìm kiếm tri thức
  - Bước 5: Đánh giá mẫu tìm được
  - Bước 6: Biểu diễn tri thức
- Ở bài này, nhóm em tóm tắt các bước thành những mục sau:
  i. Thu thập dữ liệu
  ii. Tiền xử lý dữ liệu
  iii. Biến đổi dữ liệu
  iv. Phân tách dữ liệu

3. Phân tích dữ liệu thô

- Nguồn dữ liệu thô: **UCI Machine Learning Repository: Car Evaluation Data Set**
- Hiểu dữ liệu: Dữ liệu đánh giá mức độ chất lượng của ô tô. Phân loại độ an toàn dựa trên giá trị các trọng số đánh giá
- Dữ liệu gồm: dữ liệu bao gồm 1694 bản ghi cùng 9 thuộc tính đánh giá chất lượng ô tô

	A	B	C	D	E	F	G	H	I	J
1	**order**	**buying**	**maint**	**doors**	**persons**	**lug_boot**	**safety**	**color**	**seats**	**evaluation**
2	1	vhigh	vhigh	2	2	small	low	3	2	unacc
3	2	vhigh	vhigh	2	2	small	med	4	2	unacc
4	3	vhigh	med	2	2	small	high	5	2	unacc
5	4	vhigh	vhigh	2	2	med	low	5	2	unacc
6	5	med	vhigh	2	2	med	med	5	2	unacc
7	6	vhigh	vhigh	2	2	med	high	4	2	unacc
8	7	vhigh	med	2	2	big	low	5	2	unacc
9	8	vhigh	vhigh	2	2	big	med	4	2	unacc
10	9	vhigh	vhigh	2	2	big	high	3	2	unacc
11	10	vhigh	vhigh	2	4	small	low	5	4	unacc
12	11	vhigh	vhigh	2	4	small	low	4	4	unacc
13	12	vhigh	med	2	4	small	high	5	4	unacc
14	13	vhigh	vhigh	2	4	med	low	3	4	unacc
15	14	vhigh	vhigh	2	4	med	med	5	4	unacc
16	15	vhigh	vhigh	2	4	med	high	4	4	unacc
17	16	vhigh	vhigh	2	4	big	low	3	4	unacc
18	17	vhigh	vhigh	2	5	big	med	4	5	unacc
19	18	vhigh	vhigh	2	4	big	high	3	4	unacc
20	19	med	vhigh	2	5	small	low	5	5	unacc
21	20	vhigh	vhigh	2	5	med	med	4	5	unacc
22	21	vhigh	vhigh	2	5	small	high	3	5	unacc
23	22	vhigh	vhigh	2	5	med	low	3	5	unacc
24	23	vhigh	vhigh	2	5	med	med	5	5	unacc
25	24	vhigh	med	2	5	med	high	3	5	unacc
26	25	vhigh	vhigh	2	5	big	low	3	5	unacc
27	26	vhigh	vhigh	2	5	big	med	3	5	unacc
28	27	vhigh	vhigh	2	5	big	low	3	5	unacc
29	28	vhigh	vhigh	3	2	small	low	5	2	unacc
30	29	vhigh	vhigh	3	2	small	med	5	2	unacc
31	30	med	vhigh	3	2	small	high	5	2	unacc
32	31	vhigh	vhigh	3	2	med	low	4	2	unacc
33	32	vhigh	vhigh	3	2	med	med	5	2	unacc
34	33	vhigh	med	3	2	big	low	3	2	unacc
35	34	vhigh	vhigh	3	2	big	med	5	2	unacc
36	35	vhigh	vhigh	3	2	med	high	5	2	unacc

Hình 1: Dữ liệu thô

- Hiểu các thuộc tính:

STT	Thuộc tính	Ý nghĩa thuộc tính
1	Order	Số thứ tự
2	Buying	Loại giá bán của ô tô
3	Maint	Loại giá bảo dưỡng của ô tô
4	Doors	Số cửa của ô tô
5	Persons	Số người tối đa ngồi trong ô tô
6	Lug_boot	Mức kích thước cốp của ô tô
7	Safety	Độ an toàn
8	Color	Màu sơn của ô tô
9	Seats	Số ghế ngồi của ô tô
8	Evaluation	Phân loại xe ô tô Các lớp phân loại: + acc: chấp nhận + unacc: không chấp nhận + good: tốt + vgood: rất tốt

II. Quy trình khai phá dữ liệu

1. Tiền xử lý dữ liệu

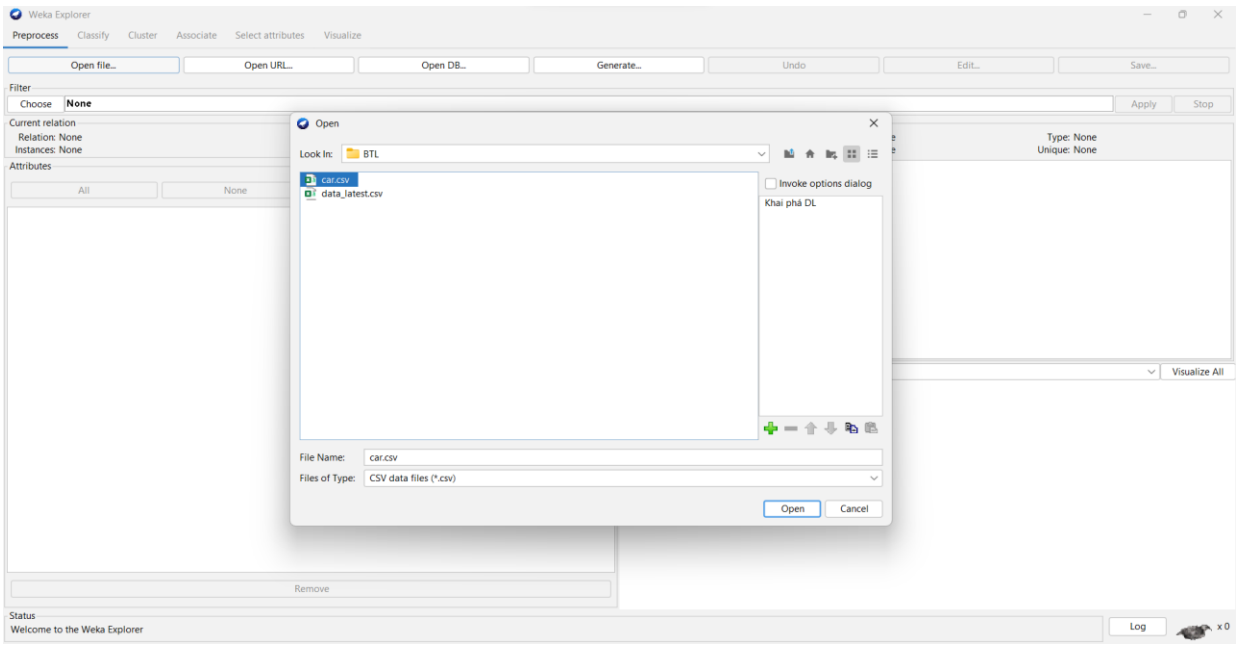
Là quá trình xử lý dữ liệu thô/gốc nhằm cải thiện chất lượng dữ liệu và chất lượng của kết quả KPDL

1.1. Làm sạch dữ liệu

Là quá trình xử lý dữ liệu bị thiếu, nhận diện phần tử biên và giảm thiểu nhiễu, xử lý dữ liệu không nhất quán

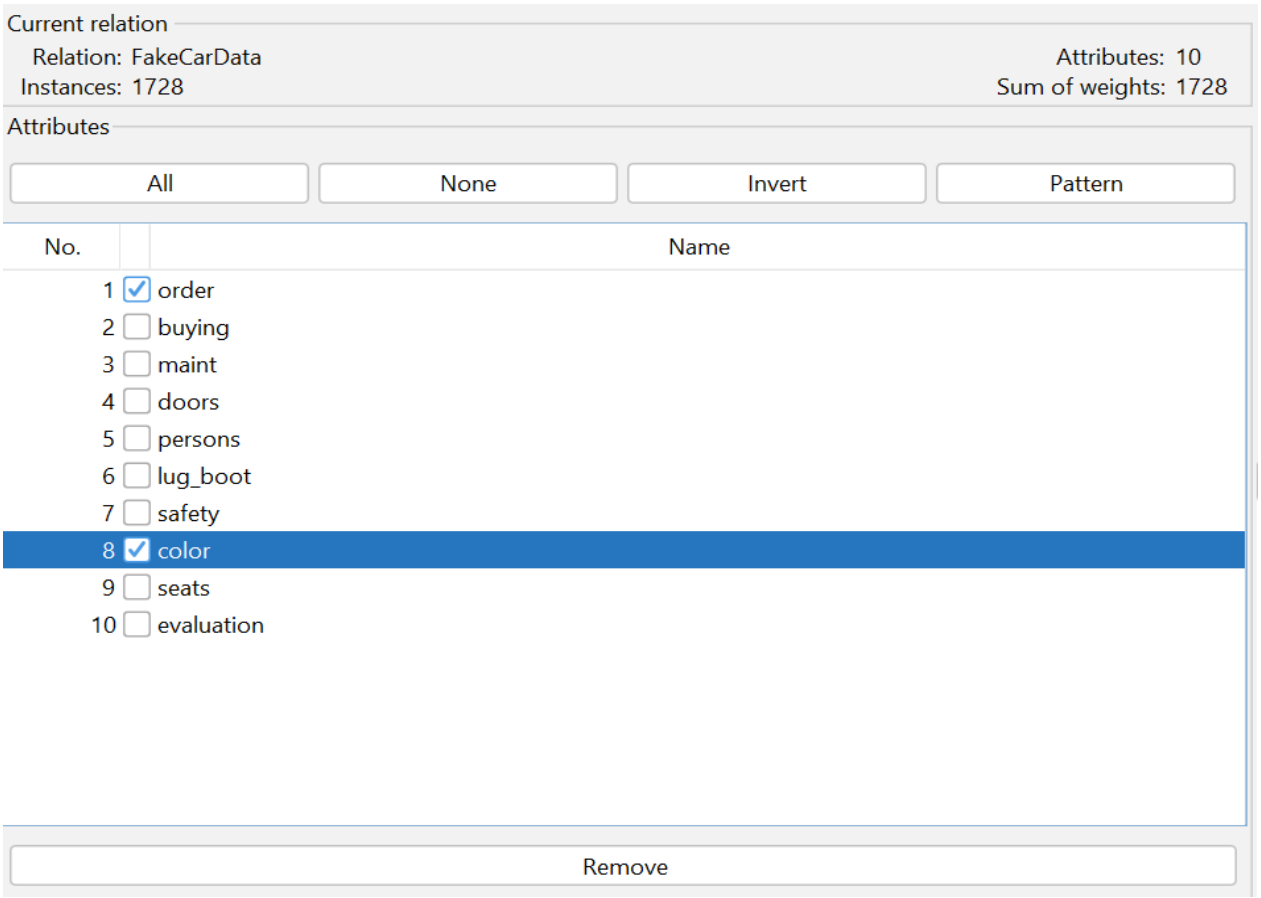
1.1.1. Loại bỏ thuộc tính dư thừa

- Đọc dữ liệu vào Weka:



Hình 2: Chọn file dữ liệu

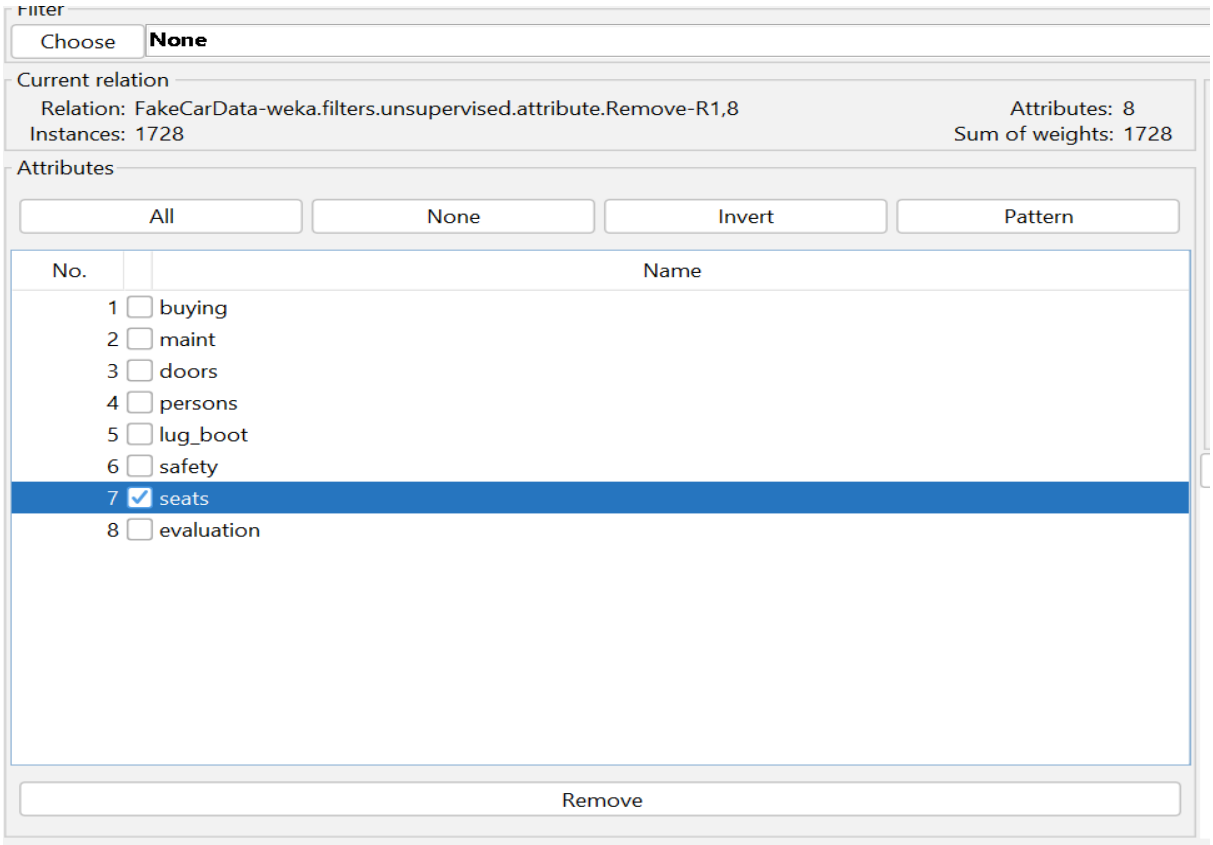
- Loại bỏ thuộc tính dư thừa, không cần thiết: **order(số thứ tự), color(màu)**
  + Tick chọn thuộc tính **order** và **color** nhấn **Remove** để loại bỏ:



Hình 3: Xóa bỏ thuộc tính Order và Color

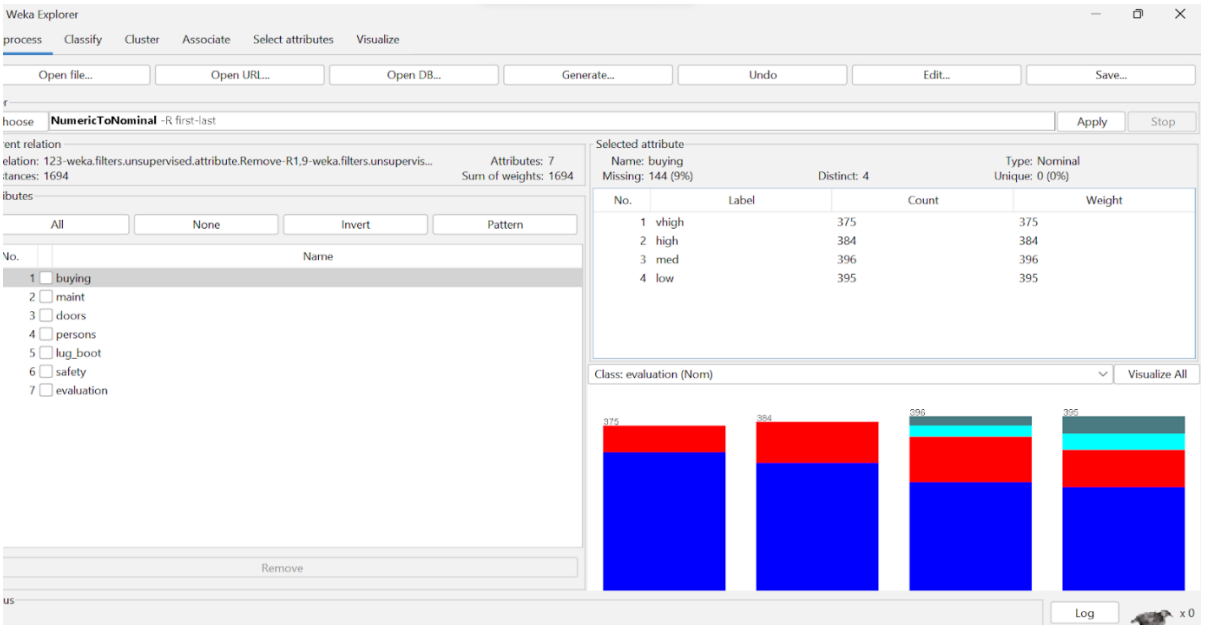
- Loại bỏ dữ liệu bị trùng lặp: dữ liệu tại thuộc tính **seats (số ghế)** trùng dữ liệu với thuộc tính **persons (số người)**

  + Tick chọn thuộc tính **seats** nhấn **Remove** để loại bỏ:



Hình 4: Xóa bỏ thuộc tính Seats

⇨ Ta thu được kết quả:



Hình 5: Dữ liệu sau khi loại bỏ thuộc tính dư thừa

1.1.2. Loại bỏ nhiễu

- Xử lý dữ liệu nhiễu, không hợp lý có thể ảnh hưởng đến kết quả phân tích:

Để thuận tiện cho việc xử lý dữ liệu bị nhiễu và phù hợp với mục tiêu khai phá, ta chuyển kiểu dữ liệu của các thuộc tính về dạng nominal:

+ **Filter -> Unsupervised -> Attribute -> NumericToNominal -> Apply**

⇨ Ta thu được kết quả:

Viewer

Relation: 123-weka.filters.unsupervised.attribute.Remove-R1,9-weka.filters.unsupervised.attribute.NumericToNominal-Rfirst-last-weka.filters.unsupervised.attribute.Remove-R7

No.	1: buying Nominal	2: maint Nominal	3: doors Nominal	4: persons Nominal	5: lug_boot Nominal	6: safety Nominal	7: **evaluation** Nominal
1	vhigh	vhigh	2	2	small	low	unacc
2	vhigh	vhigh	2	2	small	med	unacc
3	vhigh		2	2	small	high	unacc
4	vhigh	vhigh	2	2	med	low	unacc
5		vhigh	2	2		med	unacc
6	vhigh	vhigh	2	2	med	high	unacc
7	vhigh		2	2	big	low	unacc
8	vhigh	vhigh	2	2	big	med	unacc
9	vhigh	vhigh	2	2	big	high	unacc
10	vhigh	vhigh	2	4	small	low	unacc
11	vhigh	vhigh	2	4	small		unacc
12	vhigh		2	4	small	high	unacc
13	vhigh	vhigh	2	4	med	low	unacc
14	vhigh	vhigh	2	4	med	med	unacc
15	vhigh	vhigh	2	4	med	high	unacc
16	vhigh	vhigh	2	4	big	low	unacc
17	vhigh	vhigh	2		big	med	unacc
18	vhigh	vhigh	2	4	big	high	unacc
19		vhigh	2	5	small	low	unacc
20	vhigh	vhigh	2	5		med	unacc
21	vhigh	vhigh	2	5	small	high	unacc
22	vhigh	vhigh	2	5	med	low	unacc
23	vhigh	vhigh	2	5	med	med	unacc
24	vhigh		2	5	med	high	unacc

Add instance | Undo | OK | Cancel

Hình 6: Dữ liệu sau khi chuyển về dạng Nominal

- Tại thuộc tính **doors**, **persons** có một vài dữ liệu gây nhiễu như có ô tô có hơn 5 cửa (doors), 30 người (persons)… là vô lý, cần phải xử lý: **Filter -> Unsupervised -> Instance -> RemoveWithValues**

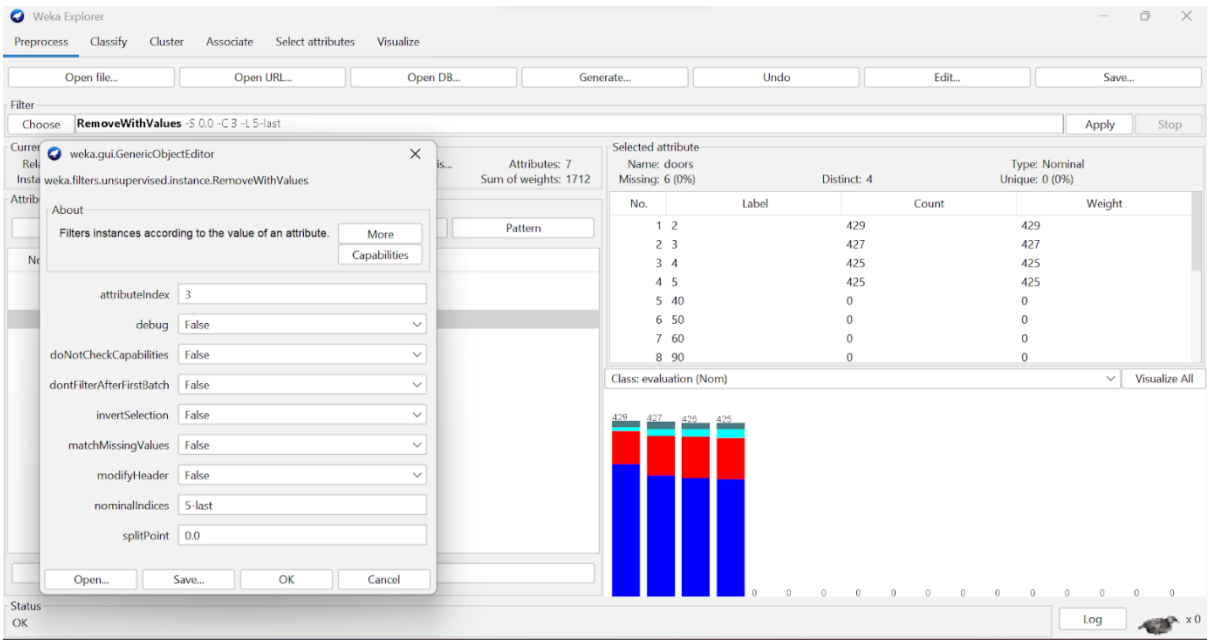
❖ <u>Với thuộc tính Doors:</u>

Selected attribute

Name: doors | Type: Nominal
Missing: 6 (0%) | Distinct: 17 | Unique: 10 (1%)

No.	Label	Count	Weight
5	40	1	1
6	50	1	1
7	60	1	1
8	90	1	1
9	100	1	1
10	200	1	1
11	300	1	1
12	400	2	2

Hình 7: Dữ liệu nhiễu tại thuộc tính Doors



Hình 8: Xóa bỏ dữ liệu nhiễu tại thuộc tính Doors

Selected attribute

Name: doors | Missing: 6 (0%) | Distinct: 4 | Type: Nominal | Unique: 0 (0%)

No.	Label	Count	Weight
1	2	429	429
2	3	427	427
3	4	425	425
4	5	425	425
5	40	0	0
6	50	0	0
7	60	0	0
8	90	0	0

Hình 9: Dữ liệu nhiễu đã xóa tại thuộc tính Doors

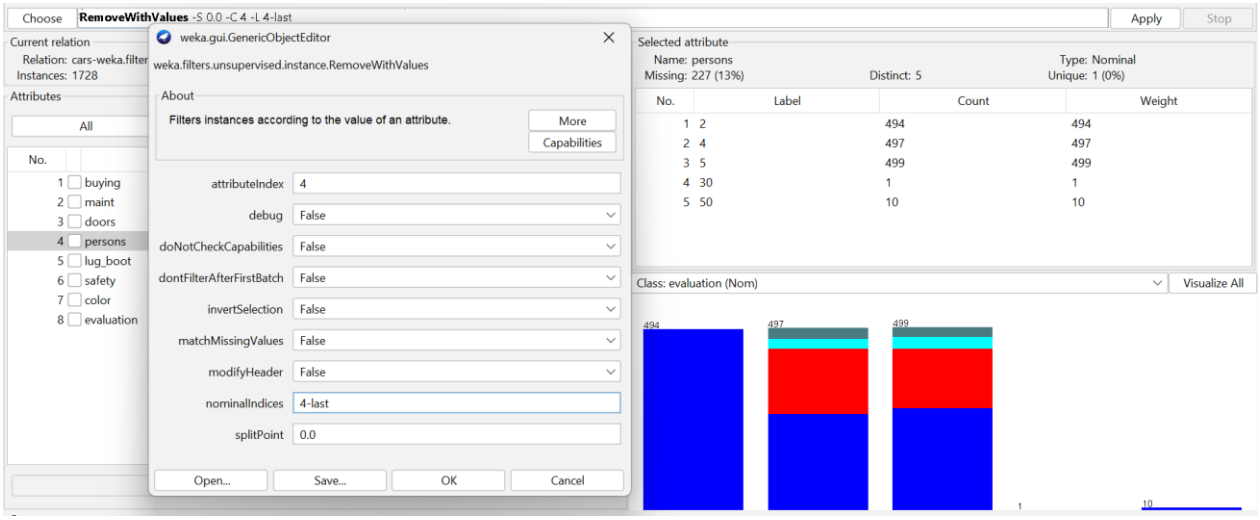
❖ <u>Với thuộc tính Persons:</u>

Selected attribute

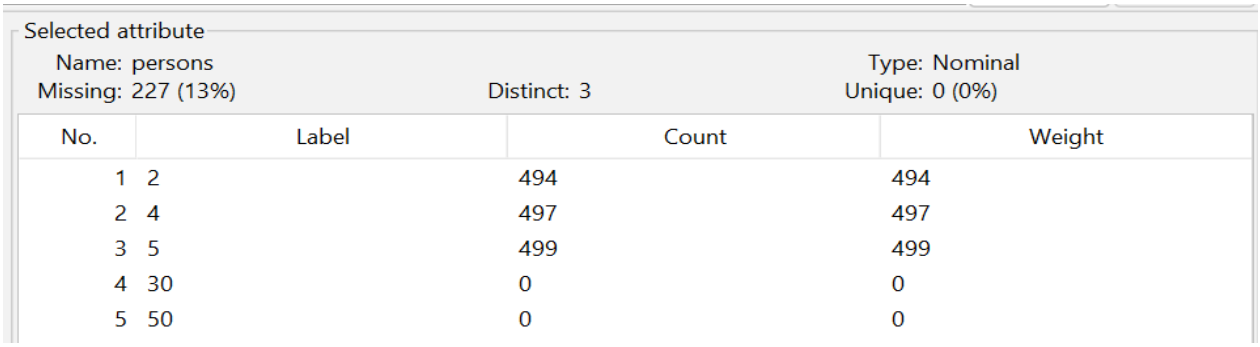
Name: persons | Missing: 44 (3%) | Distinct: 5 | Type: Nominal | Unique: 1 (0%)

No.	Label	Count	Weight
1	2	554	554
2	4	556	556
3	5	562	562
4	30	1	1
5	50	11	11

Hình 10: Dữ liệu nhiễu tại thuộc tính Persons



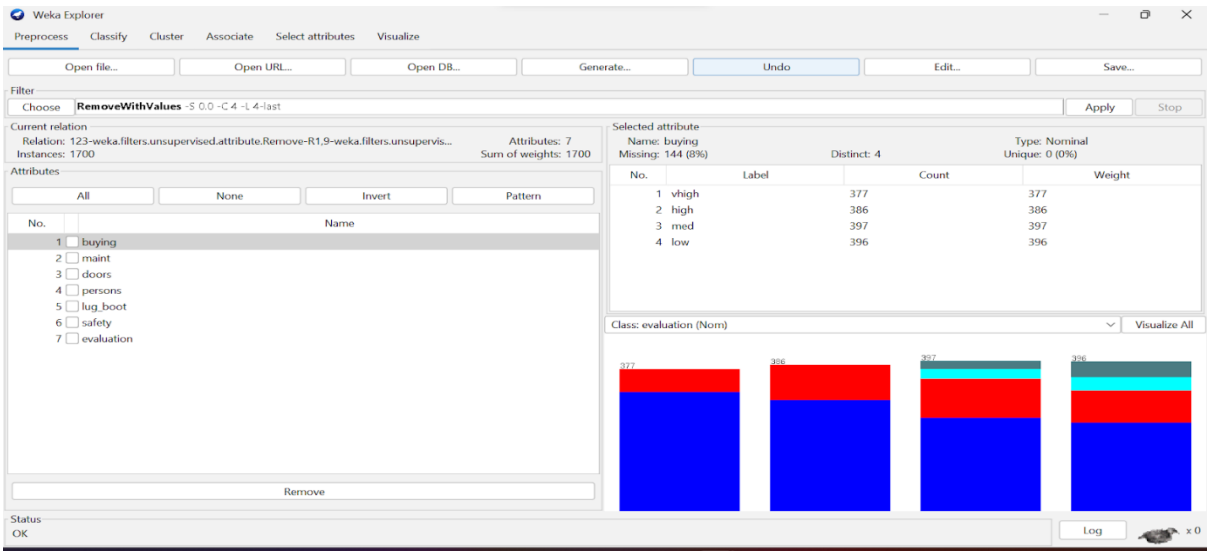
Hình 11: Xóa bỏ dữ liệu nhiễu tại thuộc tính Persons



Hình 12: Dữ liệu nhiễu đã được xóa tại thuộc tính Persons

1.1.3. Xử lý dữ liệu bị thiếu

Ví dụ: Ta có thể thấy thuộc tính **buying** bị thiếu **144** bản ghi:



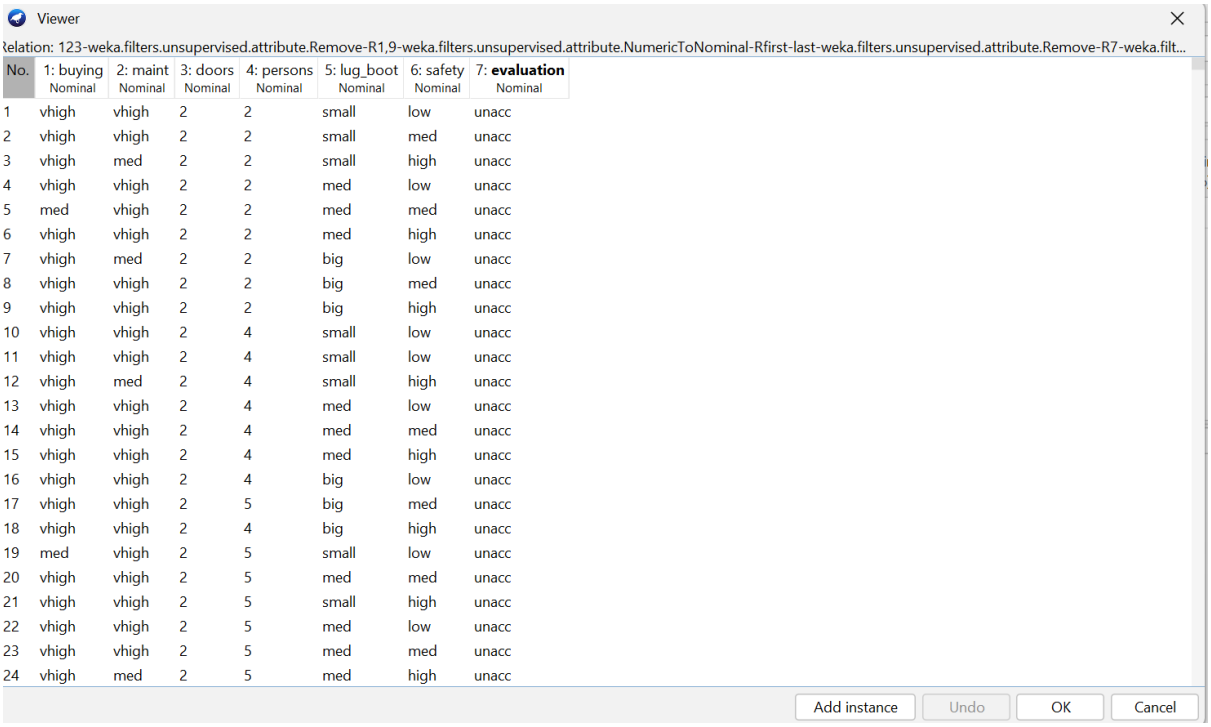
Hình 13: Thuộc tính Buying bị missing (8%)

- Thay thế các dữ liệu bị thiếu tại các thuộc tính:

Filter -> unsupervised -> attribute -> ReplaceMissingValue.

Với các dữ liệu bị thiếu dạng nominal sẽ được thay thế bằng giá trị xuất hiện nhiều nhất tại thuộc tính đó.

⇨ Dữ liệu sau khi xử lý:



No.	1: buying Nominal	2: maint Nominal	3: doors Nominal	4: persons Nominal	5: lug_boot Nominal	6: safety Nominal	7: **evaluation** Nominal
1	vhigh	vhigh	2	2	small	low	unacc
2	vhigh	vhigh	2	2	small	med	unacc
3	vhigh	med	2	2	small	high	unacc
4	vhigh	vhigh	2	2	med	low	unacc
5	med	vhigh	2	2	med	med	unacc
6	vhigh	vhigh	2	2	med	high	unacc
7	vhigh	med	2	2	big	low	unacc
8	vhigh	vhigh	2	2	big	med	unacc
9	vhigh	vhigh	2	2	big	high	unacc
10	vhigh	vhigh	2	4	small	low	unacc
11	vhigh	vhigh	2	4	small	low	unacc
12	vhigh	med	2	4	small	high	unacc
13	vhigh	vhigh	2	4	med	low	unacc
14	vhigh	vhigh	2	4	med	med	unacc
15	vhigh	vhigh	2	4	med	high	unacc
16	vhigh	vhigh	2	4	big	low	unacc
17	vhigh	vhigh	2	5	big	med	unacc
18	vhigh	vhigh	2	4	big	high	unacc
19	med	vhigh	2	5	small	low	unacc
20	vhigh	vhigh	2	5	med	med	unacc
21	vhigh	vhigh	2	5	small	high	unacc
22	vhigh	vhigh	2	5	med	low	unacc
23	vhigh	vhigh	2	5	med	med	unacc
24	vhigh	med	2	5	med	high	unacc

Hình 14: Dữ liệu sau khi xử lý các dữ liệu bị thiếu

1.1.4. Xử lý dữ liệu không nhất quán

Tập dữ liệu đã nhất quán

1.2. Tích hợp dữ liệu

- Là quá trình trộn dữ liệu từ các nguồn khác nhau vào một kho dữ liệu sẵn sàng cho quá trình khai phá dữ liệu
- Liên quan đến cấu trúc và tính không thuần nhất (heterogeneity) về ngữ nghĩa (semantics) của dữ liệu
- Hỗ trợ việc giảm và tránh dư thừa và không nhất quán về dữ liệu > cải thiện tính chính xác và tốc độ quá trình khai phá dữ liệu

⇨ **Do dữ liệu lấy từ 1 nguồn nên quy trình này bỏ qua**

1.3. Biến đổi dữ liệu (chuẩn hóa dữ liệu)

- Là quá trình biến đổi hay kết hợp dữ liệu vào những dạng thích hợp cho quá trình khai phá dữ liệu

- Bao gồm:
  - Làm trơn dữ liệu
  - Kết hợp dữ liệu
  - Tổng quát hóa
  - Chuẩn hóa
  - Xây dựng thuộc tính/đặc tính
  - Thu giảm dữ liệu

⇨ Chuyển dữ liệu của các thuộc tính về dạng nominal

2. Phân tích dữ liệu

Phân tích dữ liệu nhằm hiểu rõ hơn về dữ liệu và mối quan hệ giữa các thuộc tính do đó ta cần phân tích, nhận biết thêm về sự liên kết giữa chúng

Thống kê mô tả: Dữ liệu được đưa vào khai phá gồm **7 thuộc tính, 1694 mẫu**

STT	Thuộc tính	Ý nghĩa
1	buying	Loại giá bán của ô tô
2	maint	Loại giá bảo dưỡng của ô tô
3	doors	Số cửa cổ ô tô
4	persons	Số lượng người tối đa trong ô tô
5	lug_boot	Mức kích thước cốp hành lý của ô tô
6	safety	Độ an toàn của ô tô
7	evaluation	Mức độ đánh giá của ô tô

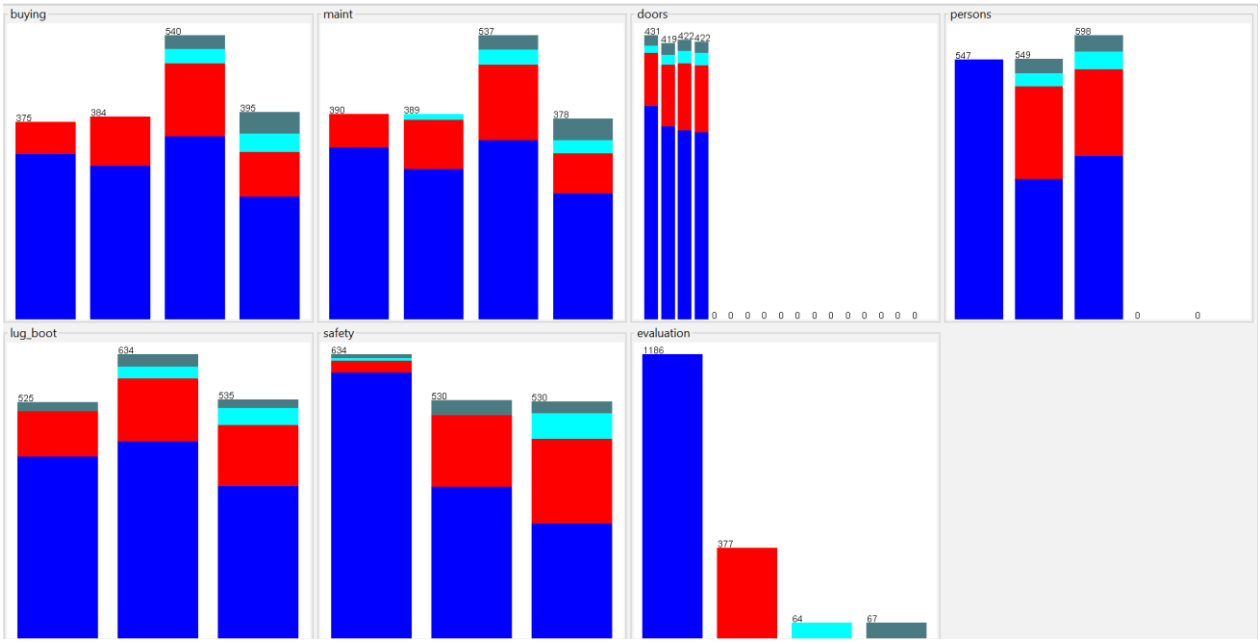
Viewer

Relation: TEST_123-weka.filters.unsupervised.attribute.Remove-R1-weka.filters.unsupervised.attribute.Remove-R7-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R7-precision7-...

No.	1: buying Nominal	2: maint Nominal	3: doors Nominal	4: persons Nominal	5: lug_boot Nominal	6: safety Nominal	7: weight Nominal	8: **evaluation** Nominal
1	vhigh	vhigh	2	2	small	low	light	unacc
2	vhigh	vhigh	2	2	small	med	light	unacc
3	vhigh	med	2	2	small	high	light	unacc
4	vhigh	vhigh	2	2	med	low	light	unacc
5	med	vhigh	2	2	med	med	light	unacc
6	vhigh	vhigh	2	2	med	high	light	unacc
7	vhigh	med	2	2	big	low	light	unacc
8	vhigh	vhigh	2	2	big	med	light	unacc
9	vhigh	vhigh	2	2	big	high	light	unacc
10	vhigh	vhigh	2	4	small	low	light	unacc
11	vhigh	vhigh	2	4	small	low	light	unacc
12	vhigh	med	2	4	small	high	light	unacc
13	vhigh	vhigh	2	4	med	low	light	unacc
14	vhigh	vhigh	2	4	med	med	light	unacc
15	vhigh	vhigh	2	4	med	high	light	unacc
16	vhigh	vhigh	2	4	big	low	light	unacc
17	vhigh	vhigh	2	5	big	med	light	unacc
18	vhigh	vhigh	2	4	big	high	light	unacc
19	med	vhigh	2	5	small	low	light	unacc
20	vhigh	vhigh	2	5	med	med	light	unacc
21	vhigh	vhigh	2	5	small	high	light	unacc
22	vhigh	vhigh	2	5	med	low	light	unacc
23	vhigh	vhigh	2	5	med	med	light	unacc
24	vhigh	med	2	5	med	high	light	unacc

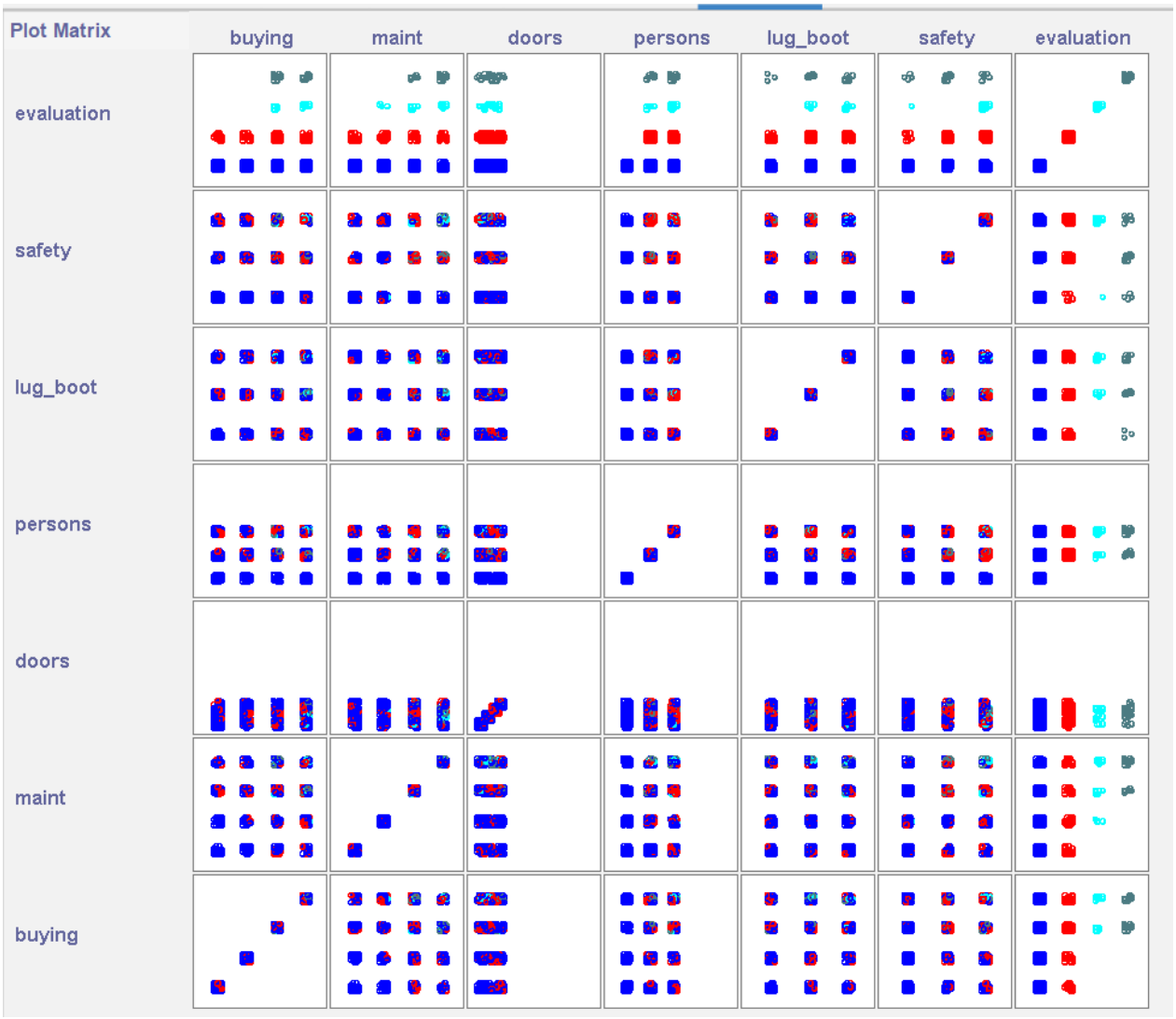
Add instance | Undo | OK | Cancel

Hình 15: Dữ liệu sau khi tiền xử lý



Hình 16: Thống kê chi tiết các thuộc tính

- Mối tương quan của dữ liệu:



Hình 11:Biểu đồ tổng quát mối quan hệ giữa các thuộc tính

Ví dụ: mối quan hệ giữa thuộc tính persons (người) và evaluation (mức đánh giá)

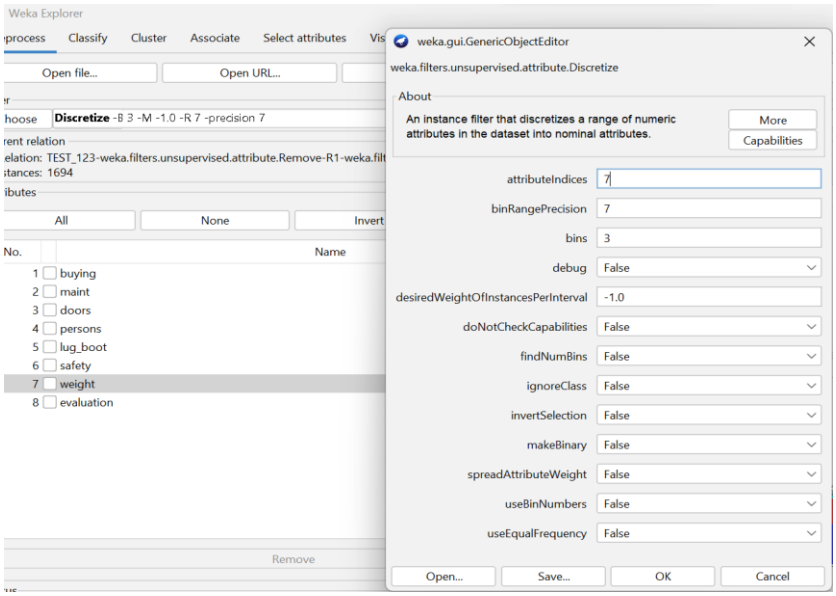
Hình 12: Biểu đồ quan hệ của thuộc tính Persons

⇨ Kết luận:

- Những xe có **2 chỗ (persons)** thì sẽ có tỉ lệ cao sẽ là **không thể chấp nhận (unacc)**
- Những xe có **4, 5 chỗ (persons)** sẽ có kết quả dự đoán **chấp nhận (acc)**, **thích (good), rất thích (vgood)** với số lượng rất lớn

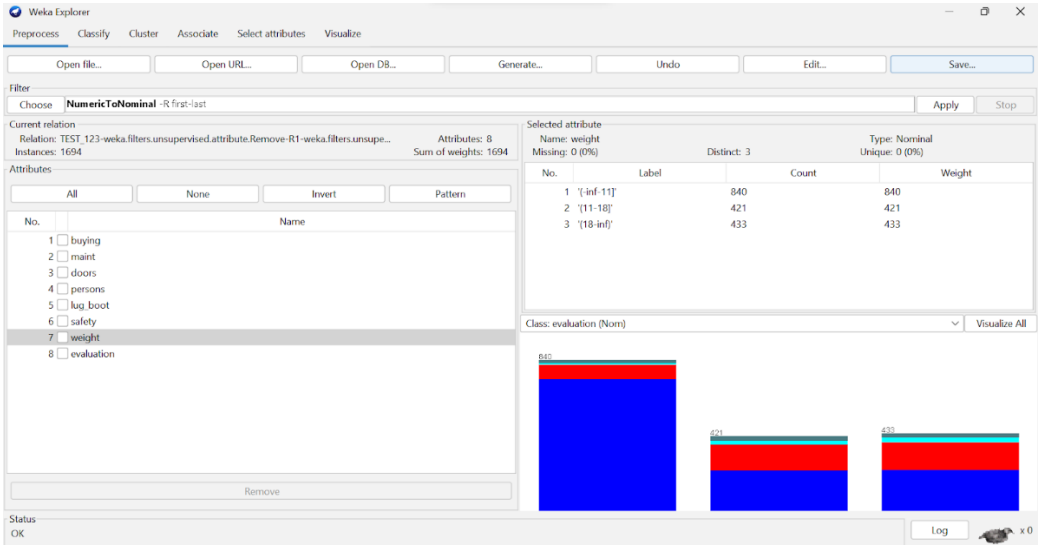
3. Biến đổi dữ liệu

- Dựa vào dữ liệu đã được xử lý ta có thể xây dựng thuộc tính **weight = person * door,** để thể hiện sự liên kết giữa các thuộc tính, đem lại tính tích cực trong việc khai phá dữ liệu, xây dựng mô hình đánh giá
- Chia bins: bộ lọc Discretize, ta chia 3 bins



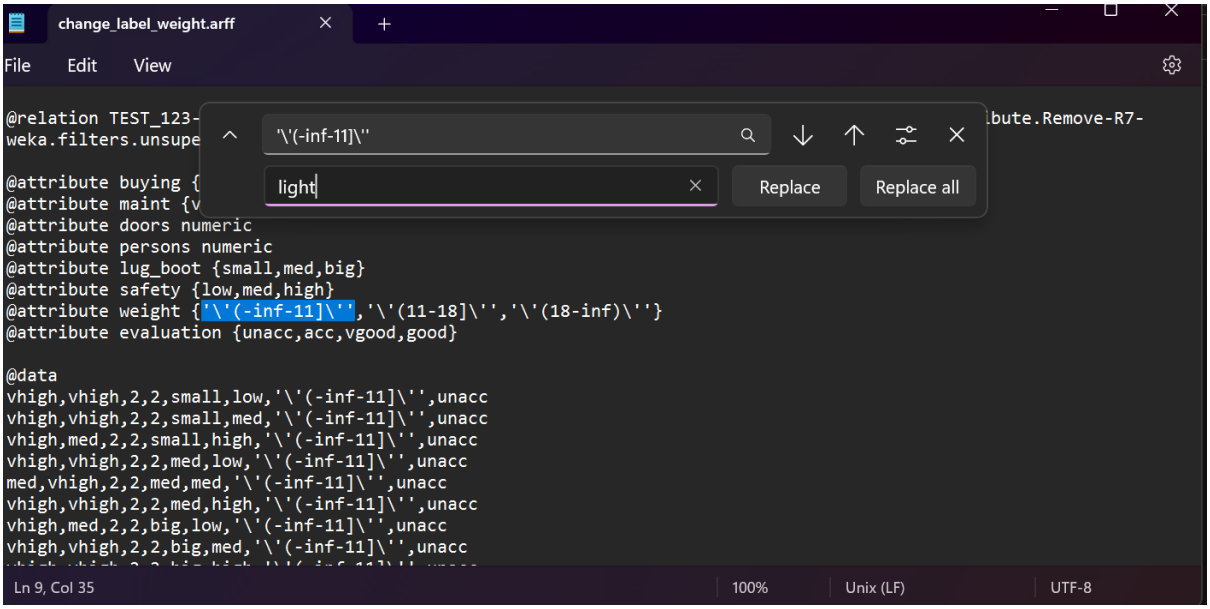
Hình 13: Quy trình tạo thuộc tính Weight

⇨ Ta thu được kết quả:



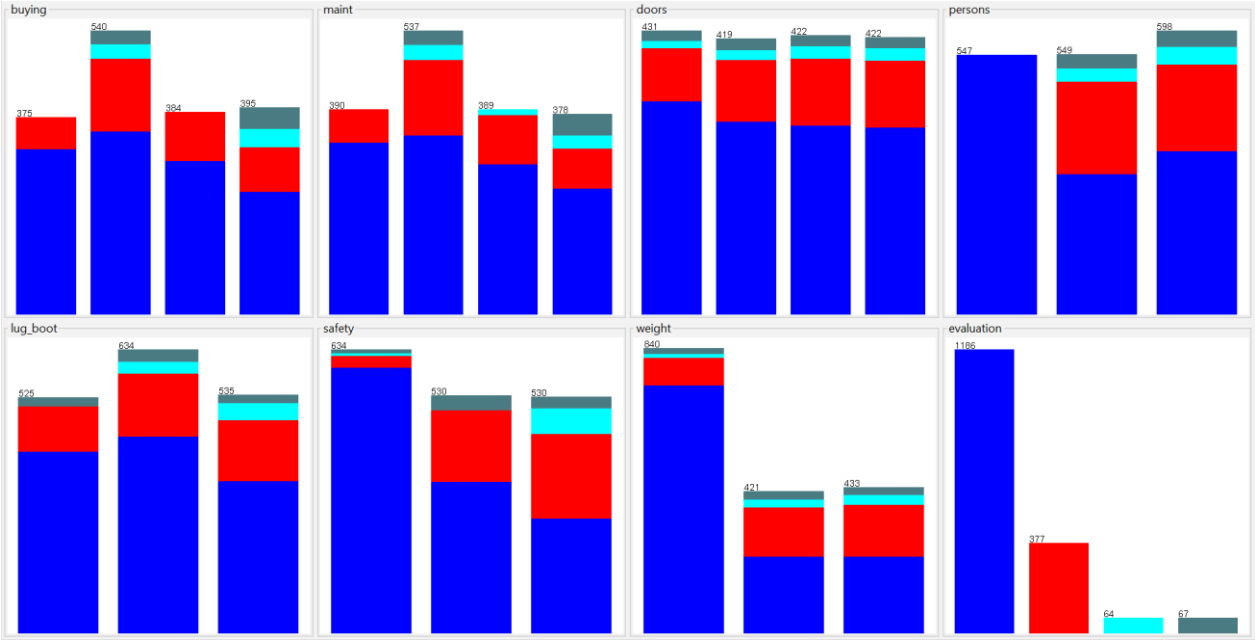
Hình 20: Thuộc tính Weight sai khi được chia làm 3 miền

- Tiếp tục ta chỉnh sửa dữ liệu: gán các nhãn với từng khoảng dữ liệu được chia đều tương ứng, giúp thuận tiện hơn việc hiểu dữ liệu
  - [4 , 11] = light
  - [12, 18] = med
  - [19, 25] = heavy



Hình 21: Thay đổi label của thuộc tính Weight

⇨ Ta thu được kết quả:



Hình 22: Kết quả dữ liệu

- Miền giá trị của các thuộc tính:

STT	Thuộc tính	Miền giá trị
1	Buying	{vhigh, high, med, low}
2	Maint	{vhigh, high, med, low}
3	Doors	{2, 4, 5}
4	Persons	{2, 3, 4, 5}
5	Lug_boot	{small, med, big}
6	Safety	{high, med, low}
7	Weight	{light,med,heavy}
8	Evaluation	{unacc, acc, good, vgood}

4. Phân tách dữ liệu (70/30)

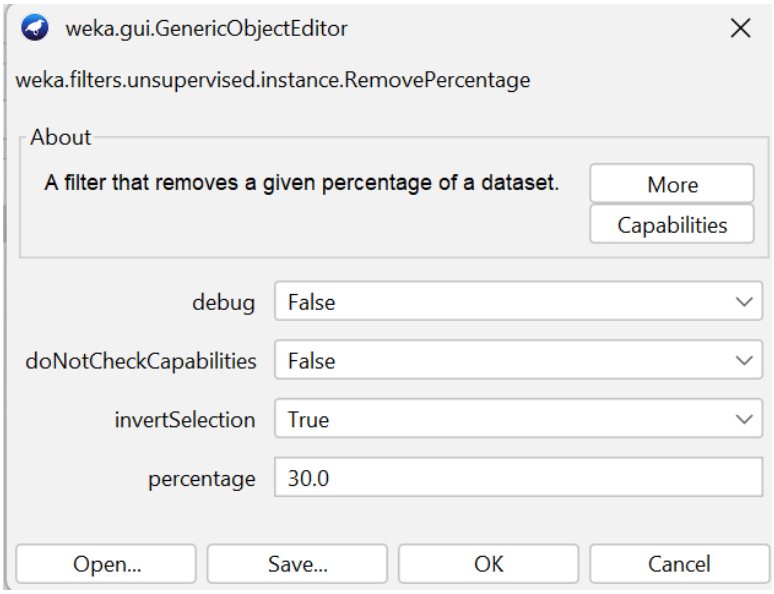
- Lý do:
  - Phân chia dữ liệu huấn luyện và kiểm thử trong phân lớp giúp đánh giá hiệu suất của mô hình trên tập dữ liệu mới. Dữ liệu huấn luyện được sử dụng để huấn luyện mô hình và dữ liệu kiểm thử được sử dụng để đánh giá hiệu suất của mô hình trên tập dữ liệu mới.
  - Phân chia dữ liệu huấn luyện và kiểm thử cũng giúp tránh tình trạng overfitting
- Dữ liệu huấn luyện và dữ liệu kiểm thử
  - Dữ liệu huấn luyện (Data training): là tập dữ liệu được sử dụng để xây dựng thuật toán, mô hình học máy

- Dữ liệu kiểm thử (Data test): là tập dữ liệu được đưa vào sau khi đã xây dựng xong thuật toán, mô hình học máy, nhằm xác nhận tính hiệu quả của mô hình đó và mang tính thực tế

- Lựa chọn tỉ lệ:
  - Để khách quan trong việc xây dựng mô hình ta trộn dữ liệu ngẫu nhiên**:**

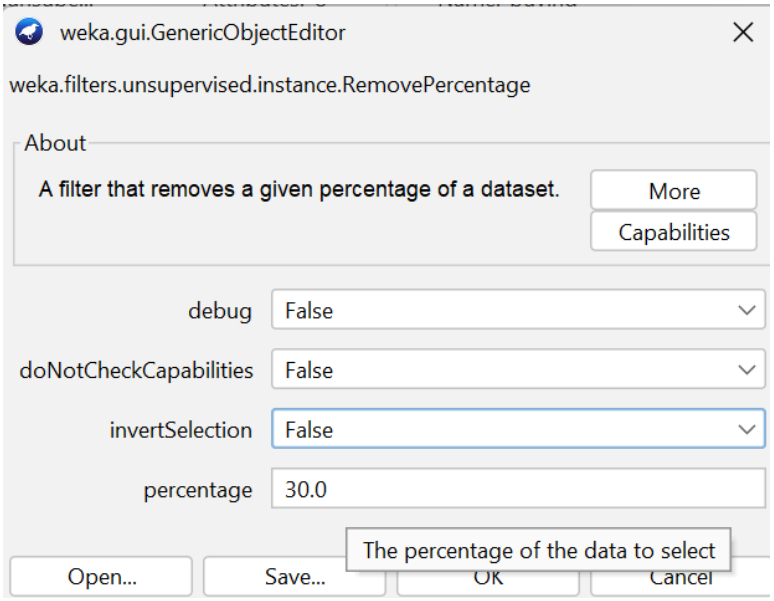
  Filter -> unsupervised -> instance -> Randomize
  - Sử dụng bộ lọc **RemovePercentage** để chia tập dữ liệu huấn luyện và kiểm thử.

- Dữ liệu kiểm thử:



Hình 23: Phân chia dữ liệu 30% là tập Test

- Dữ liệu huấn luyện:



Hình 24: Phân chia dữ liệu 70% là tập Train

⇨ Kết quả: Sau khi chọn tỉ lệ, ta xáo trộn dữ liệu thu được tập dữ liệu huấn luyện và tập dữ liệu kiểm thử với tỉ lệ 70/30

III. Phương pháp phân lớp và thuật toán ID3

1. Bài toán phân lớp

- Phân lớp dữ liệu là kĩ thuật dựa trên tập huấn luyện, những giá trị hay nhãn dán của lớp trong một thuộc tính phân lớp và sử dụng nó trong việc phân lớp dữ liệu mới.
- Phân lớp là một hình thức học được giám sát: dữ liệu mới được phân lớp dựa trên tập huấn luyện.
- Quá trình phân lớp dữ liệu gồm hai bước:
  - Xây dựng mô hình
  - Sử dụng mô hình

2. Thuật toán ID3

2.1. Lý thuyết

- Là một thuật toán học các mẫu để tạo cây quyết định. Tạo cây quyết định bằng việc tìm kiếm cơ bản từ trên xuống trên tập huấn luyện
- Thuật toán ID3 bắt đầu với một tập dữ liệu ban đầu và một tập các thuộc tính có thể được sử dụng để phân loại. Sử dụng độ lợi thông tin để tìm ra thuộc tính tốt nhất chia các mẫu thành các nhóm. Thuật toán tiếp tục phân tách các nhóm con đến khi tất cả các mẫu trong mỗi nhóm đều thuộc cùng một lớp hoặc không còn thuộc tính nào có thể được sử dụng để phân loại.
- Khi cây quyết định được xây dựng, việc phân loại của một mẫu mới được thực hiện bằng cách đi xuống các nhánh của cây theo các giá trị của các thuộc tính cho đến khi đạt tới một lá cây. Lá này sẽ chỉ ra lớp mà mẫu đó thuộc về.
- Hạn chế: Bị overfitting nếu cây quá phức tạp và khả năng bị ảnh hưởng bởi các thuộc tính có số lượng giá trị lớn.
- Độ lợi thông tin:

  - Thông tin mong đợi để phân lớp một mẫu trong D theo nhãn lớp:

$$Entropy(D) = -\sum_{i=1}^{m} P_i \log_2(P_i)$$

  - Thông tin cần thiết để phân chia D theo thuộc tính A:

$$Entropy_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} Entropy(D_j)$$

  - Độ lợi thông tin của sự phận chia dựa trên thuộc tính A:

$$Gain(A) = Entropy(D) - Entropy_A(D)$$

2.2. Quy trình thực hiện

Bước 1: Xác định thuộc tính chính để phân loại đối tượng.

Bước 2: Tạo một nút trên cây quyết định dựa trên thuộc tính đã chọn

Bước 3: Chia tập dữ liệu thành các tập con dựa trên giá trị của thuộc tính đã chọn ở bước 1

Bước 4: Lặp lại các bước trên cho tập dữ liệu con được chọn trong bước 3

Bước 5: Dừng lại khi một trong các điều kiện sau được thỏa mãn:

- Tất cả các đối tượng trong tập dữ liệu đang xét đều thuộc cùng một lớp.
- Không còn thuộc tính nào để chọn.
- Số lượng đối tượng trong tập dữ liệu đang xét quá nhỏ hoặc không đủ để chia thành các tập con.

Bước 6: Gắn nhãn cây với lớp được phổ biến nhất của các đối tượng trong tập dữ liệu đã xét

2.3. Kết quả thu được

Test options với Use training set:

```
Correctly Classified Instances        1139               96.0371 %
Incorrectly Classified Instances        47                3.9629 %
Kappa statistic                          0.9102
Mean absolute error                      0.0208
Root mean squared error                  0.102
Relative absolute error                  9.0662 %
Root relative squared error             30.1423 %
Total Number of Instances             1186

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.999    0.097    0.959      0.999    0.979      0.928    0.998     0.998     unacc
                 0.901    0.013    0.954      0.901    0.927      0.906    0.997     0.984     acc
                 0.727    0.000    1.000      0.727    0.842      0.848    0.998     0.946     vgood
                 0.829    0.000    1.000      0.829    0.907      0.908    0.999     0.975     good
Weighted Avg.    0.960    0.071    0.961      0.960    0.959      0.920    0.998     0.992

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 826   1   0   0 |   a = unacc
  27 247   0   0 |   b = acc
   5   7  32   0 |   c = vgood
   3   4   0  34 |   d = good
```

Hình 25: Khi sử dụng Use training set

- **Test options với Cross-validation:**

```
Correctly Classified Instances         923               77.8246 %
Incorrectly Classified Instances       155               13.0691 %
Kappa statistic                          0.6514
Mean absolute error                      0.0753
Root mean squared error                  0.2673
Relative absolute error                 37.6357 %
Root relative squared error             86.486  %
UnClassified Instances                 108                9.1062 %
Total Number of Instances             1186

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.934    0.219    0.921      0.934    0.928      0.724    0.869     0.906     unacc
                 0.682    0.074    0.707      0.682    0.694      0.616    0.752     0.502     acc
                 0.529    0.011    0.600      0.529    0.563      0.550    0.697     0.267     vgood
                 0.484    0.016    0.469      0.484    0.476      0.460    0.674     0.193     good
Weighted Avg.    0.856    0.176    0.854      0.856    0.855      0.689    0.834     0.782

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 738  46   2   4 |   a = unacc
  54 152   7  10 |   b = acc
   5   8  18   3 |   c = vgood
   4   9   3  15 |   d = good
```

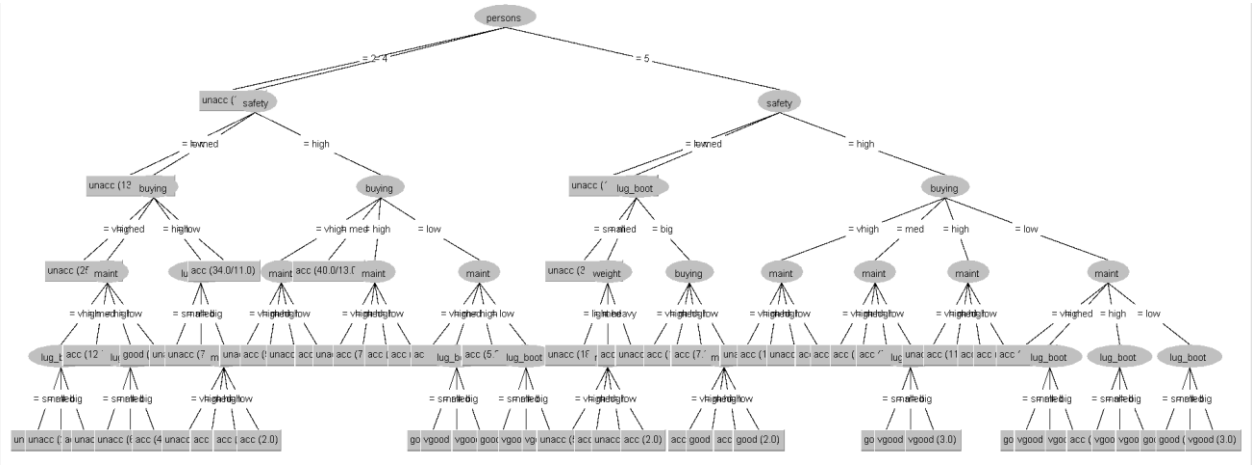
Hình 26: Khi sử dụng Cross-vadation

- **Test options với Supplied test set:**

```
Correctly Classified Instances         457               89.9606 %
Incorrectly Classified Instances        36                7.0866 %
Kappa statistic                          0.8369
Mean absolute error                      0.0369
Root mean squared error                  0.1688
Relative absolute error                 16.4911 %
Root relative squared error             50.3492 %
UnClassified Instances                  15                2.9528 %
Total Number of Instances              508

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.126    0.946      0.980    0.963      0.875    0.958     0.968     unacc
                 0.867    0.037    0.875      0.867    0.871      0.833    0.936     0.833     acc
                 0.600    0.002    0.900      0.600    0.720      0.728    0.896     0.687     vgood
                 0.652    0.004    0.882      0.652    0.750      0.749    0.813     0.576     good
Weighted Avg.    0.927    0.096    0.926      0.927    0.924      0.855    0.944     0.910

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 335   7   0   0 |   a = unacc
  15  98   0   0 |   b = acc
   1   3   9   2 |   c = vgood
   3   4   1  15 |   d = good
```

Hình 27: Khi sử dụng Supplied test set

- **Xây dựng cây quyết định:**



Hình 21: Cây quyết định

- Giải thích các thống kê:

- **Correctly Classified Instances:** Số lượng các mẫu trong tập dữ liệu được phân loại chính xác bởi mô hình.

- **Incorrectly Classified Instances:** Số lượng các mẫu trong tập dữ liệu bị phân loại sai bởi mô hình .

- **Kappa statistic (**Thống kê kappa): Đây là một chỉ số đo lường sự đồng ý giữa nhãn lớp được dự đoán và thực tế, được hiệu chỉnh để tính đến sự trùng hợp ngẫu nhiên.

- **Mean absolute error (**Sai số trung bình tuyệt đối): Sai số trung bình giữa xác suất dự đoán của mô hình và xác suất thực tế tương ứng với mỗi mẫu.

- **Root mean squared error** (Căn bậc hai của sai số trung bình bình phương): là căn bậc hai của sai số trung bình bình phương giữa xác suất dự đoán của mô hình và xác suất thực tế tương ứng với mỗi mẫu.

- **Relative absolute error** (Sai số tuyệt đối tương đối): Đây là sai số trung bình giữa xác suất dự đoán của mô hình và xác suất thực tế, được chuẩn hóa bởi xác suất thực tế trung bình.

- **Root relative squared error** (Căn bậc hai của sai số bình phương tương đối): Đây là căn bậc hai của sai số trung bình bình phương giữa xác suất dự đoán của mô hình và xác suất thực tế, được chuẩn hóa bởi xác suất thực tế trung bình.

- **Total Number of Instances:** Tổng số lượng mẫu trong tập dữ liệu

- **UnClassified Instances** (Số lượng các mẫu không phân loại): Đây là số lượng các mẫu trong tập dữ liệu kiểm tra không thể được phân loại bởi mô hình.

Giải thích các độ đo:

- **TP Rate:** Tỷ lệ số lượng các mẫu được phân loại chính xác vào nhãn Positive (dương tính) so với tổng số mẫu Positive trong tập dữ liệu

- **FP Rate:** Tỷ lệ số lượng các mẫu bị phân loại sai vào nhãn Positive so với tổng số mẫu Negative trong tập dữ liệu.

- **Precision:** Tỷ lệ số lượng các mẫu được phân loại chính xác vào nhãn Positive so với tổng số các mẫu được phân loại vào nhãn Positive.

- **Recall:** Tỷ lệ số lượng các mẫu được phân loại chính xác vào nhãn Positive so với tổng số mẫu Positive trong tập dữ liệu.

- **F-Measure:** Kết hợp giữa Precision và Recall để đánh giá hiệu quả phân loại. F-Measure càng lớn thì phân loại càng chính xác.

- **MCC**: Độ đo tính tương đồng của hai chuỗi số. MCC = 1 tương đương với việc phân loại hoàn hảo và MCC = -1 tương đương với việc phân loại hoàn toàn ngược lại.

- **ROC Area:** Đường cong ROC được sử dụng để đánh giá hiệu quả của thuật toán phân loại trong bài toán dự đoán nhị phân. ROC Area là diện tích dưới đường cong ROC.

- **PRC Area:** Đường cong Precision-Recall được sử dụng để đánh giá hiệu quả của thuật toán phân loại trong bài toán dự đoán nhị phân. PRC Area là diện tích dưới đường cong Precision-Recall

⇨ Đánh giá mô hình:

- Ta có kết luận khi sử dụng với mô hình thuật toán ID3:
  + Với cross validation : cho tỉ lệ đúng khá cao với tỉ lệ chính xác là 80%
  + Với tập dữ liệu data train để huấn luyện với dữ liệu data Test để kiểm thử cho ra tỉ lệ chính xác là 90%.

⇨ Với hai cách kiểm thử tỉ lệ trên , cho ra tỉ lệ đúng là dao động quanh 90%

- Đối với bài toán dự đoán chất lượng ô tô, việc sử dụng thuật toán ID3 là hoàn toàn phù hợp. Bởi vì dữ liệu của bài toán này là dữ liệu khá đơn giản và có mối quan hệ logic rõ ràng giữa các thuộc tính của ô tô và chất lượng của ô tô. Việc sử dụng thuật toán ID3 cho phép xây dựng một cây quyết định dễ hiểu và có thể giải thích được quá trình phân lớp của dữ liệu.

- Tuy nhiên, như các thuật toán khác, ID3 cũng có những hạn chế. Điểm yếu của thuật toán này là khi sử dụng với dữ liệu phức tạp hơn và có nhiều thuộc tính, nó có thể dẫn đến hiện tượng quá khớp (overfitting) hoặc không phát hiện được các quy tắc quan trọng trong dữ liệu. Để giải quyết vấn đề này, các biến thể của thuật toán ID3 đã được phát triển để cải thiện khả năng tổng quát hóa và độ chính xác của cây quyết định.

⇨ Tóm lại, với bài toán dự đoán chất lượng ô tô, việc sử dụng thuật toán ID3 là một lựa chọn tốt. Tuy nhiên, để đạt được kết quả tốt nhất, ta cần chú ý đến việc giải quyết hiện tượng quá khớp và sử dụng các biến thể của thuật toán để cải thiện độ chính xác và khả năng tổng quát hóa của cây quyết định.

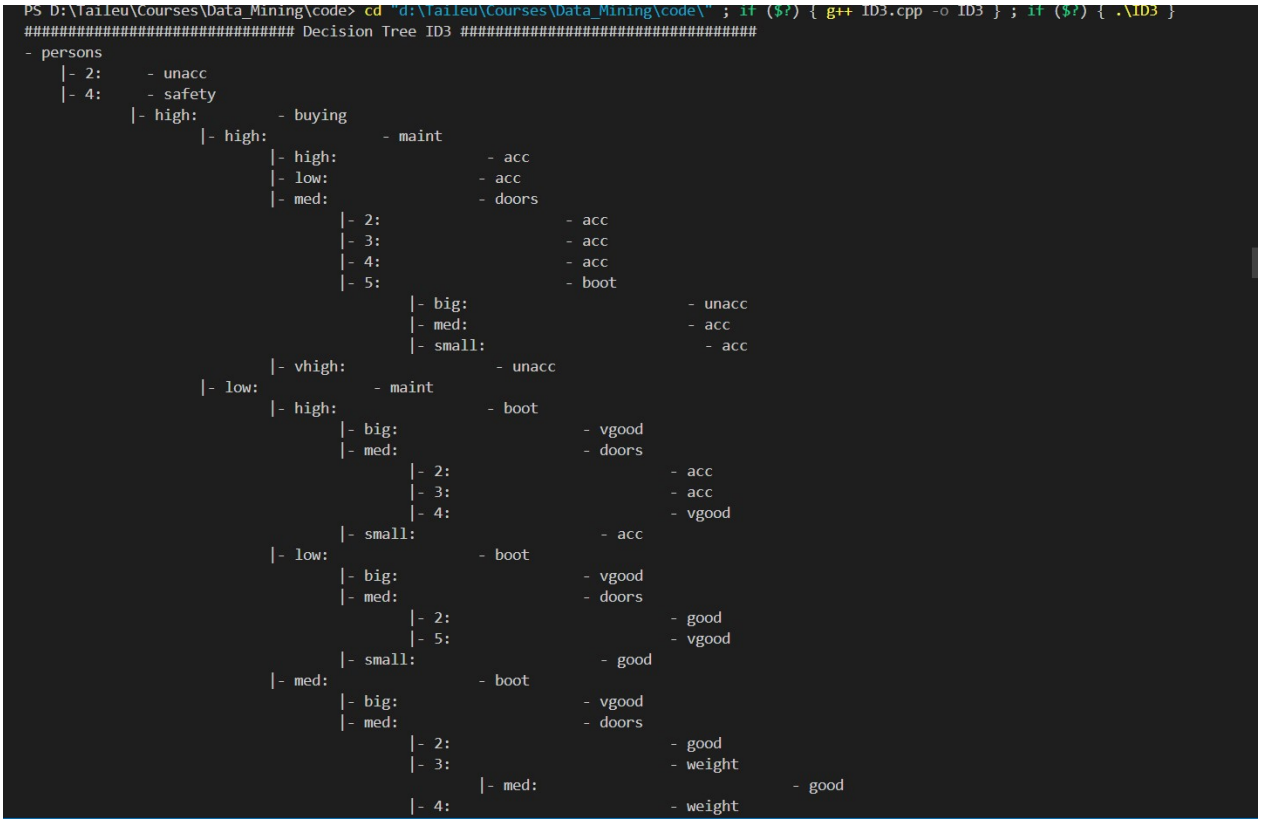
IV. Triển khai thuật toán (C++)

1. Xây dựng cây

```
Node* ID3(vector<Car>& cars, vector<string>& attributes) {

    Node* node = newNode();

    bool same_type = true;
    string type = cars[0].type;
    for (auto car : cars) {
        if (car.type != type) {
            same_type = false;
            break;
        }
    }
    if (same_type) {
        node->label = type;
        return node;
    }
    if (attributes.size() == 0) {
        node->label = type;
        return node;
    }
    string best_attribute = chooseAttribute(cars, attributes);
    node->label = best_attribute;

    vector<string> new_attributes = attributes;
    new_attributes.erase(remove(new_attributes.begin(), new_attributes.end(), best_attribute), new_attributes.end());

    map<string, vector<Car>> partitions;
    for (auto car : cars) {
            if(best_attribute==listAttribute[0])        partitions[car.buying].push_back(car);
            else if(best_attribute==listAttribute[1])   partitions[car.maint].push_back(car);
            else if(best_attribute==listAttribute[2])   partitions[car.doors].push_back(car);
            else if(best_attribute==listAttribute[3])   partitions[car.persons].push_back(car);
            else if(best_attribute==listAttribute[4])   partitions[car.boot].push_back(car);
            else if(best_attribute==listAttribute[5])   partitions[car.safety].push_back(car);
            else if(best_attribute==listAttribute[6])   partitions[car.weight].push_back(car);
    }
    for (auto partition :partitions) {
        Node* child = newNode();
        child->label = partition.first;
        vector<Car> subset = partition.second;
        if (subset.size() == 0) child->label = type;
        else child = ID3(subset, new_attributes);
        node->children[partition.first] = child;
    }
    return node;
}
```

Hình 29: Hàm ID3 để sinh ra cây quyết định

⇨ Ta thu được kết quả:



Hình 30: Kết quả cây quyết định sau khi chạy code

⇨ So sánh với kết quả chạy bằng Weka thì output in C++ cho ra kết quả đúng.

2. Sử dụng mô hình để dự đoán kết quả

- Nhập Buying
- Nhập Maint
- Nhập Person
- Nhập Doors
- Nhập Lug Boot
- Nhập Safety
- Nhập Weight

```
Car new_car ;
cout<<"Input Buying (vhigh,high,med,low) : ";
cin>>new_car.buying;
cout<<"Input Maint (vhigh,high,med,low) : ";
cin>>new_car.maint;
cout<<"Input Persons (2,4,5) : ";
cin>>new_car.persons;
cout<<"Input Doors (2,3,4,5) : ";
cin>>new_car.doors;
cout<<"Input Lug_boot (small,med,big) : ";
cin>>new_car.boot;
cout<<"Input Safety (high,med,low) : ";
cin>>new_car.safety;
cout<<"Input Weight (light,med,heavy) : ";
cin>>new_car.weight;
string predicted_label = predict(root, new_car);
cout << "Vehicle quality prediction with data => {"<<new_car.buying<<","<<new_car.maint<<","<<new_car.persons<<",
"<<new_car.doors<<","<<new_car.boot<<","<<new_car.safety<<","<<new_car.weight<<"} is : " << convert(predicted_label);
return 0;
```

Hình 31: Code nhập dữ liệu

- Sau khi nhập dữ liệu đầu vào tương ứng ở trên , mô hình sẽ sử dụng hàm dự đoán trong code (xem chi tiết ở file đính kèm) để cho ra kết quả

Nhãn	Output
uacc	Unacceptable car quality
acc	Acceptable car quality
good	Good car quality
vgood	Very good car quality

- Trường hợp dữ liệu không nằm trong phạm vi dự đoán:
- ⇨ Kết quả sẽ trả về: Unpredictable

```
############################## Predict ################################
Input Buying (vhigh,high,med,low) : nul
Input Maint (vhigh,high,med,low) : nul
Input Persons (2,4,5) : null
Input Doors (2,3,4,5) : null
Input Lug_boot (small,med,big) : null
Input Safety (high,med,low) : null
Input Weight (light,med,heavy) : null
Vehicle quality prediction with data => {nul,nul,null,null,null,null,null} is : Unpredictable !!!
-------------------------------
```

Hình 32: Kết quả sau khi nhập dữ liệu không hợp lý

- Thực hành chạy code dự đoán:

```
############################## Predict ################################
Input Buying (vhigh,high,med,low) : high
Input Maint (vhigh,high,med,low) : vhigh
Input Persons (2,4,5) : 4
Input Doors (2,3,4,5) : 4
Input Lug_boot (small,med,big) : med
Input Safety (high,med,low) : high
Input Weight (light,med,heavy) : med
unacc
Vehicle quality prediction with data => {high,vhigh,4,4,med,high,med} is : Unacceptable car quality
```

Hình 33: Kết qua sau khi nhập dữ liệu hợp lý

KẾT LUẬN

Khai phá dữ liệu với phương pháp phân lớp là một trong những phương pháp phổ biến, mang tính ứng dụng, hiệu quả thực tiễn cao trong các lĩnh vực như thương mại, y tế, giáo dục,... và sẽ tiếp tục phát triển trong tương lai.

Sau khi hoàn thành đề tài "**Khai phá dữ liệu Mức độ đánh giá chất lượng ô tô con bằng phương pháp phân lớp",** nhóm em đã đạt được kết quả sau:

- Tìm hiểu tổng quan về khám phá tri thức, khai phá dữ liệu, bài toán phân lớp cùng thuật toán ID3 để xây dựng mô hình phân lớp hỗ trợ dự đoán mức độ đánh giá chất lượng ô tô con.

- Thu thập dữ liệu đánh giá, tiền xử lý dữ liệu với các công cụ phần mềm như Excel, Weka. Xây dựng được mô hình phân lớp trên Weka.

- So sánh kết quả mô hình phân lớp với cả phương pháp kiểm thử khác nhau để thu được mô hình tốt nhất.

Trong quá trình hoàn thành đề tài, dù nhóm em đã cố gắng tìm hiểu và thực hiện bài toán nhưng vẫn không thể tránh những thiếu sót, khiếm khuyết. Do vậy, rất mong nhận được sự đóng góp ý kiến của thầy cô để chúng em rút kinh nghiệm, cải thiện kỹ năng và tích lũy kiến thức trong môn học.

TÀI LIỆU THAM KHẢO

[1] TS.Đặng Thị Thu Hiền, Nguyễn Tu Trung, Bài giảng Khai phá dữ liệu

[2] https://archive.ics.uci.edu/ml/datasets/car+evaluation

[3] https://machinelearningcoban.com/2018/01/14/id3/