ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO LAB 1 – TIỀN XỬ LÝ VÀ KHÁM PHÁ DỮ LIỆU

MÔN HỌC: KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

LỚP: 20_21

NĂM HỌC: 2022-2023

GIẢNG VIÊN LÝ THUYẾT: GS.TS. LÊ HOÀI BẮC

GIẢNG VIÊN THỰC HÀNH: NGUYỄN THỊ THU HẰNG

MỤC LỤC

I. GIỚI THIỆU CHUNG:

1. Thông tin nhóm:

Nhóm gồm 2 thành viên:

- Nguyễn Duy Hưng – 20120096
- Lương Văn Triều – 20120604

2. Bảng phân công công việc:

Thành viên thực hiện	**Công việc**	**Độ hoàn thành**
Nguyễn Duy Hưng	Install WEKA, Getting Acquainted With WEKA	**100%**
Lương Văn Triều	Preprocessing Data in Python	**100%**

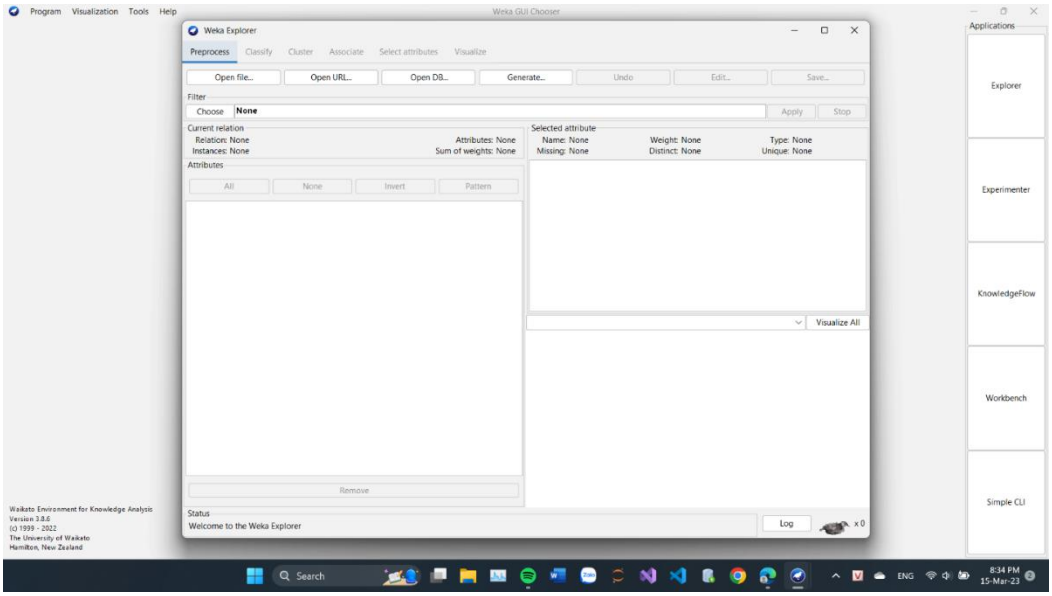
3. Đánh giá mức độ hoàn thành:

Đánh giá mức độ hoàn thành trên toàn bài tập: 100%

II. CHI TIẾT BÁO CÁO:

1. Install WEKA

– *Requirement 1:*



– *Requirement 2:*

+ Current Relation: hiển thị tên của cơ sở dữ liệu vừa mở và cho ta biết được tên của tập dữ liệu, số lượng thuộc tính, số trường hợp, tổng trọng số

+ Attributes: hiển thị các trường khác nhau trong cơ sở dữ liệu

+ Selected Attribute:

- Hiển thị tên và loại thuộc tính được hiển thị
- Loại của thuộc tính
- Số lượng giá trị “Missing”
- Số lượng dòng dữ liệu riêng biệt không có giá trị duy nhất
- Số lượng dòng dữ liệu duy nhất

+ Các tab trong Explorer của weka

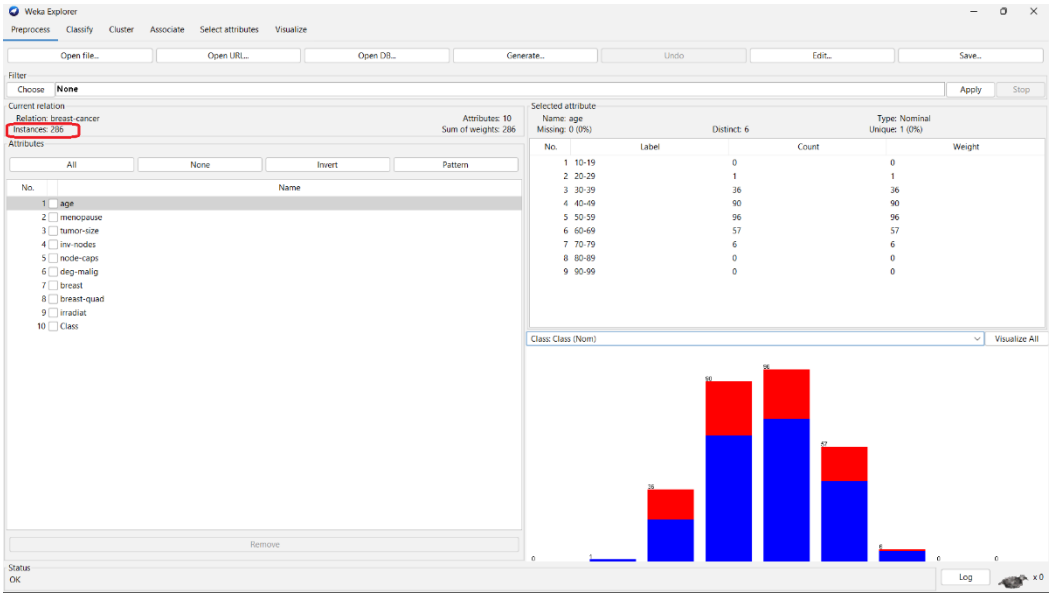
- Preprocessing: Hiển thị thông số ban đầu chưa qua xử lý
- Classify: Phân lớp dữ liệu theo các mô hình
- Cluster: Phân cụm dữ liệu theo các mô hình
- Assosiate: Khám phá các luật kết hợp của dữ liệu
- Select attriube: Quyết định các thuộc tính tương quan
- Visuallize: Biểu diễn trực quan dữ liệu

2. Getting Acquainted with WEKA

2.1 Exploring Breast Cancer data set

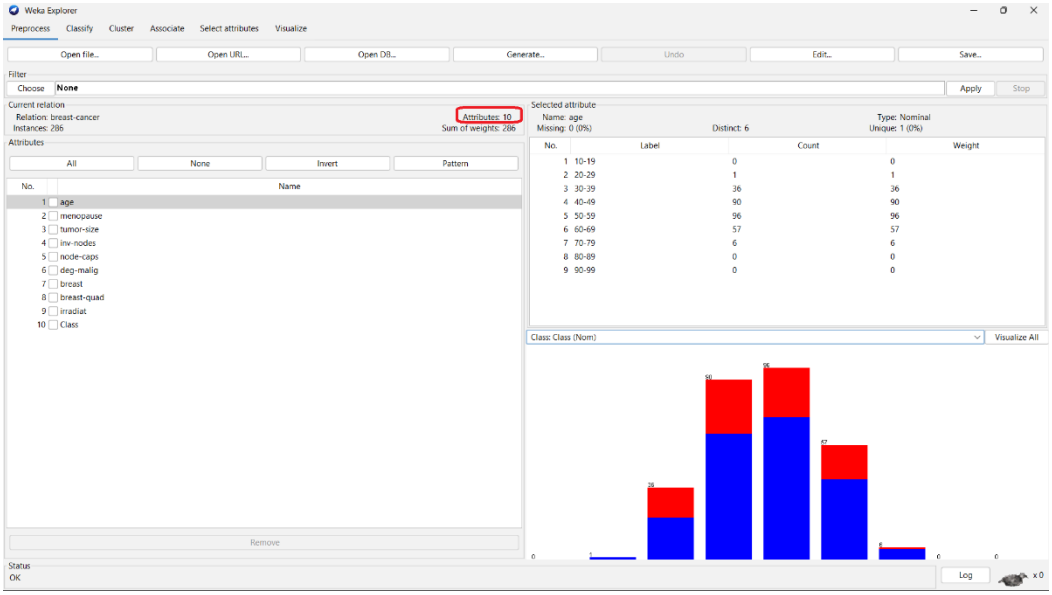
– *How many instances does this data set have?*

Tập dữ liệu có 286 trường hợp(instances)



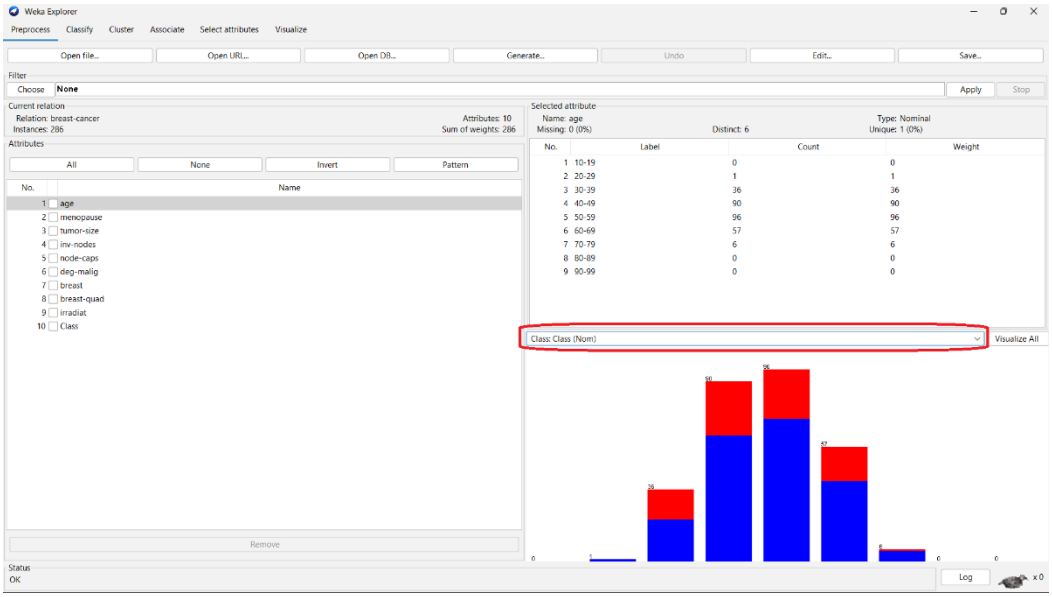
– *How many attributes does this data set have?*

Tập dữ liệu có 10 thuộc tính(attributes)



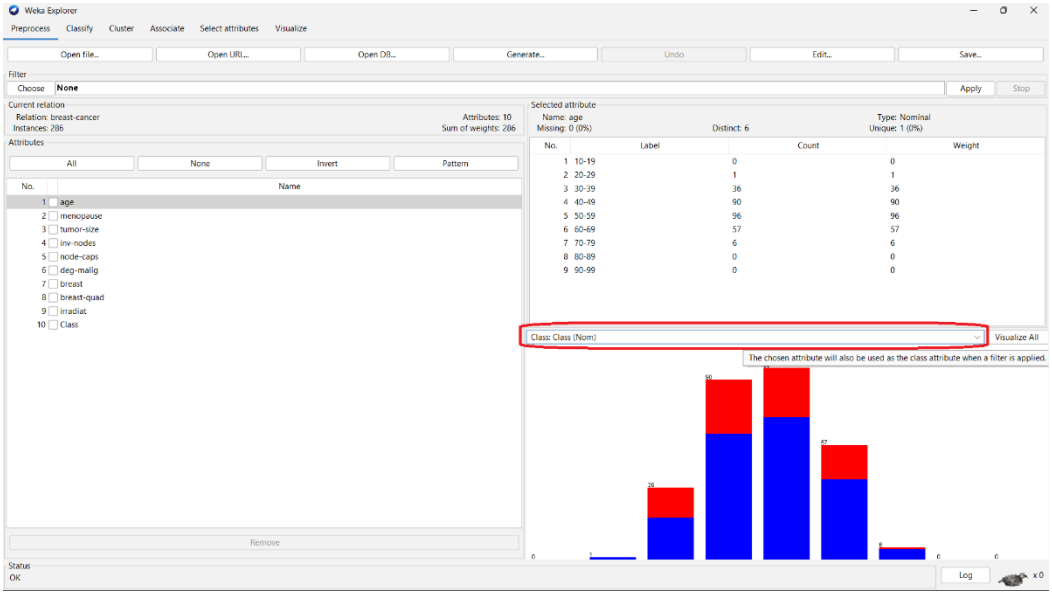
– *Which attribute is used for the label? Can it be changed? How?*

Attribute used for the label is **Class**, ta có thể thay đổi được

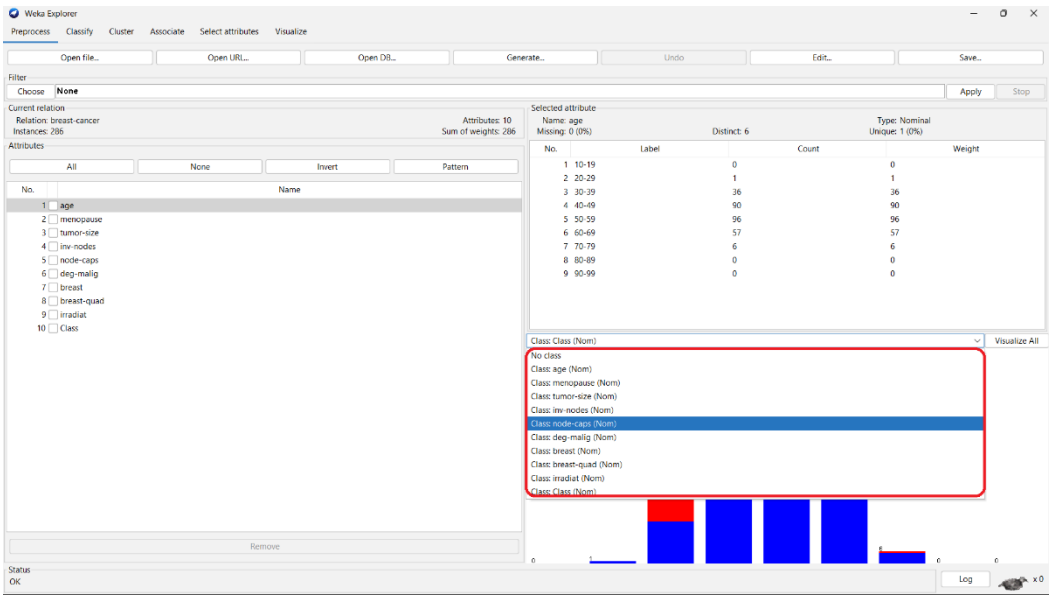


Ta có thể thay đổi được bằng cách làm theo hình bên dưới:

1. Click vào **Class: Class (Nom)**



2. Sau đó có thể chọn các label như mong muốn:



– *What is the meaning of each attribute?*

- Age: tuổi của bệnh nhân

- Menopause: cho biết số lượng bệnh nhân trước mãn kinh và sau mãn kinh
- Tumor size: kích thước khối u
- Inv-nodes: số lượng của các hạch bạch huyết nách có chứa ung thư vú di căn có thể nhìn thấy khi kiểm tra mô học
- Node-caps: cho biết khối u có thẻ xâm nhập vào viên nang và xâm lấn các mô hay không
- Deg-malig: mức độ ác tính
- Breast: số lượng vú ung thư bên trái, phải
- Breast-quad: các phần của vú
- Irradiat: có thể xạ trị được hay không
- Class: không tái phát và tái phát

– Let's investigate the missing value status in each attribute and describe in general ways to solve the problem of missing values.

Cách xử lý tình trạng missing values:

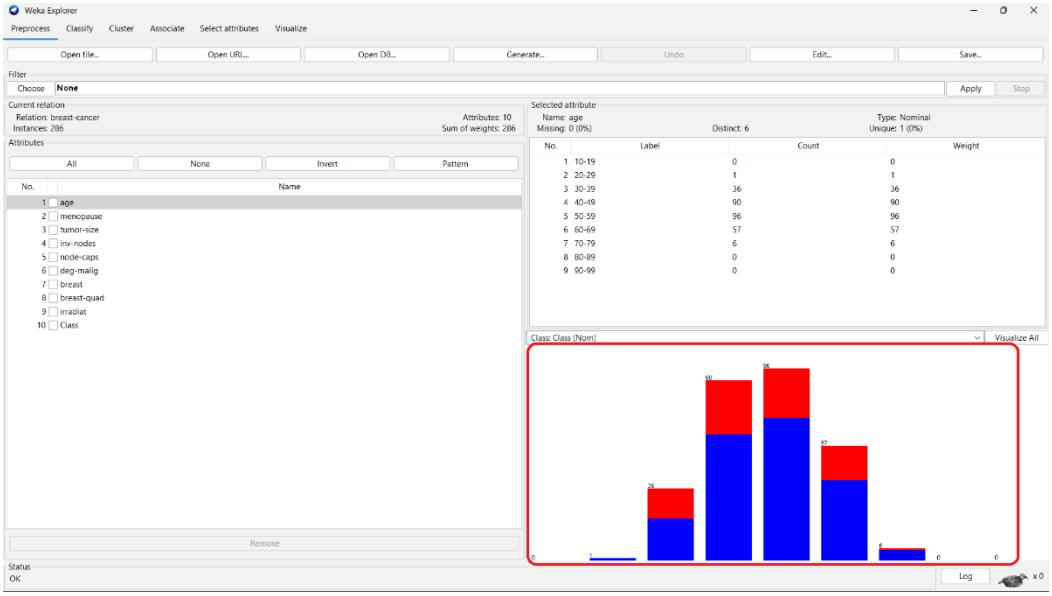
- Loại bỏ missing values (trong trường hợp missing values đó không quan trọng đối với dữ liệu của chúng ta hoặc số lượng missing values quá ít - chỉ chiếm khoảng dưới 3% tổng số quan sát trong 1 biến nhất định)
- Thay thế missing values bằng một giá trị khác. Việc thay thế bằng giá trị nào sẽ phụ thuộc vào việc bản chất của missing values trong những trường hợp đó là gì (thay thế bằng giá trị mean, midian, mode,…)

– Let's propose solutions to the problem of missing values in the specific attribute.

- Node-caps: loại bỏ missing values
- Breast-quad: loại bỏ missing values

– Let's explain the meaning of the chart in the WEKA Explorer. Setting the title for it and describing its legend.

Biểu đồ trong WEKA Explorer cho biết giá trị, so sánh các đại lượng trong 1 thuộc tính và tỉ lệ giữa tái phát với không tái phát trong từng đại lượng của thuộc tính. Tên của biểu đồ: Biểu đồ thể hiện giá trị và tỉ lệ tái phát-không tái phát của thuộc tính.



2.2 Exploring Weather data set

– *How many attributes does this data set have? How many samples? Which attributes have data type categorical? Which attributes have a data type that is numerical? Which attribute is used for the label?*

- Tập dữ liệu có 5 thuộc tính (attributes)
- Tập dữ liệu có 14 mẫu (samples)
- Thuộc tính nào có kiểu dữ liệu phân loại (categorical): Outlook, windy, play
- Thuộc tính nào có kiểu dữ liệu là số (numerical): Temperature, humidity
- Attribute used for the label is **Play**

- Let's list five-number summary of two attributes temperature and humidity. Does WEKA provide these values?

Five-number	Temperature	Humidity
Min	64	65
Tứ phân vị Q1(25%)	69.25	71.25
Median	72	82.5
Tứ phân vị Q3(25%)	78.75	90
Max	85	96

WEKA có cung cấp những giá trị này (ngoại trừ hai tứ phân vị)

– Let's explain the meaning of all charts in the WEKA Explorer. Setting the title for it and describing its legend.

- Outlook: cho biết số lượng thời tiết nắng, âm u, mưa và tỉ lệ giữa đi chơi và không đi chơi trong mỗi đại lượng thời tiết
- Temperature: cho biết số lượng của từng khoảng nhiệt độ thì tỉ lệ đi và không đi chơi như thế nào trong mỗi khoảng nhiệt độ
- Humidity: cho biết số lượng của từng khoảng độ ẩm thì tỉ lệ đi và không đi chơi như thế nào trong mỗi khoảng độ ẩm
- Windy: cho biết số lượng của trời có gió, không có gió và tỉ lệ đi và không đi chơi trong lúc có gió và không có gió
- Play: cho biết số lượng của đi và không đi chơi

– Let's move to the Visualize tag. What's the name of this chart? Do you think there are any pairs of different attributes that have correlated?

- Tên của biểu đồ này là: plot matrix
- Không có bất kỳ cặp thuộc tính nào có tương quan(correlated)

2.3 Exploring Credit in Germany data set

– *What is the content of the comments section in credit-g.arff (when opened with any text editor) about? How many samples does the data set have? How many attributes? Describe any five attributes (must have both discrete and continuous attributes).*

+ Nội dung của phần nhận xét là mô tả bộ dữ liệu tín dụng của đức, tên file, số lượng mẫu, kiểu dữ liệu từng thuộc tính

+ Tập dữ liệu có 1000 mẫu (samples)

+ Tập dữ liệu có 21 thuộc tính (attributes)

+ Describe any five attributes (must have both discrete and continuous attributes):

1. Checking_status

Kiểu dữ liệu: chuỗi

Thuộc tính: không liên tục

Mô tả trạng thái tài khoản hiện tại chính là lương trong ít nhất 1 năm

2. credit_history

Kiểu dữ liệu: chuỗi

Thuộc tính: không liên tục

Mô tả lịch sử tín dụng

3. credit_amount

Kiểu dữ liệu: số

Thuộc tính: liên tục

Cho biết số tiền tín dụng

4. property_magnitude

Kiểu dữ liệu: chuỗi

Thuộc tính: không liên tục

Cho biết loại tài sản gồm: bất động sản, thỏa thuận tiết kiệm xã hội xây dựng hoặc bảo hiểm nhân thọ, xe, không xách định

5. Age

Kiểu dữ liệu: chuỗi

Thuộc tính: không liên tục

Tuổi theo năm

– *Which attribute is used for the label?*

Attribute used for the label is: class

– Let's describe the distribution of continuous attributes? (Left skewed or right skewed?)

- Thuộc tính duration: Left skewed
- Thuộc tính credit_amount: Left skewed
- Thuộc tính age: Left skewed

– *Let's explain the meaning of all charts in the WEKA Explorer. Setting the title for it and describing its legend.*

Trong biểu đồ của các thuộc tính 1-20 đều có tỉ lệ giữa tốt và xấu trong các cột giá trị

1. checking_status: trạng thái của tài khoản kiểm tra hiện tại

2. duration: Thời lượng trong tháng

3. credit_history: Lịch sử tín dụng

4. purpose: Mục đích

5. credit_amount: số tiền tín dụng

6. savings_status: Tài khoản tiết kiệm/trái phiếu

7. employment: việc làm hiện tại kể từ

8. installment_commitment: Tỷ lệ trả góp tính theo phần trăm thu nhập khả dụng

9. personal_status: Tình trạng cá nhân và giới tính

10. other_parties: Con nợ/người bảo lãnh khác

11. residence_since: cư trú hiện tại kể từ

12. property_magnitude: tài sản

13. age: tuổi theo năm

14. other_payment_plans: Gói trả góp khác

15. housing: nhà ở

16. existing_credits: Số lượng tín dụng hiện có tại ngân hàng này

17. job: công việc

18. num_dependents: Số người chịu trách nhiệm cung cấp bảo trì cho

19. own_telephone: chủ của số điện thoại

20. foreign_worker: lao động nước ngoài

21. class:

– Let's move to the Select attributes tag. Describe all of the options for attribute selection.

+ Trong **Attribute Evaluator** sẽ có các options sau:

- CfsSubsetEval
- GainRatioAttributeEval

- InfoGainAttributeEval
- OneRAttributeEvalOneRAttributeEval
- PrincipalComponents
- ReliefFAttributeEval
- SymmetricalUncertAttributeEval
- WrapperSubsetEval

+ Trong **Search Method** sẽ có các options sau:

- BestFirst
- GreedyStepwise
- Ranker

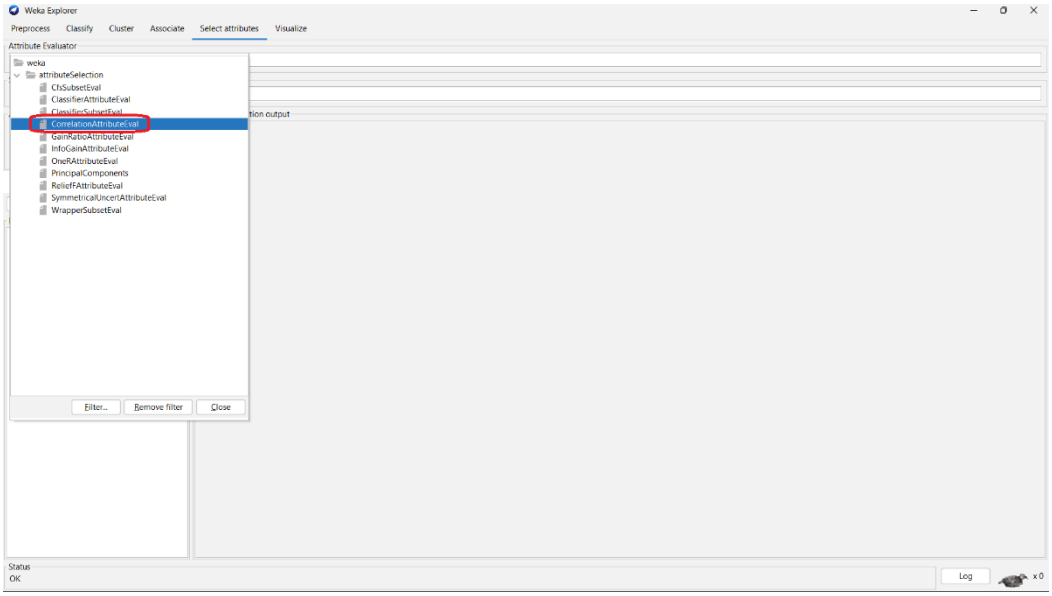
+ Trong **Attribute Selection Mode** sẽ có 2 lựa chọn đó là "Use full training set" và "Cross-vaildation".

Click vào "Start" để xử lý tập dữ liệu thì ta thấy xuất hiện thông tin bên **Attribute selection output**. Ở dưới cùng của **Attribute selection output** ta sẽ thấy được dánh sách **Selected attributes.** Để có biểu diễn trực quan thì ta click chuột phải vào kết quả trong **Result list** chọn **Visualize reduced data**

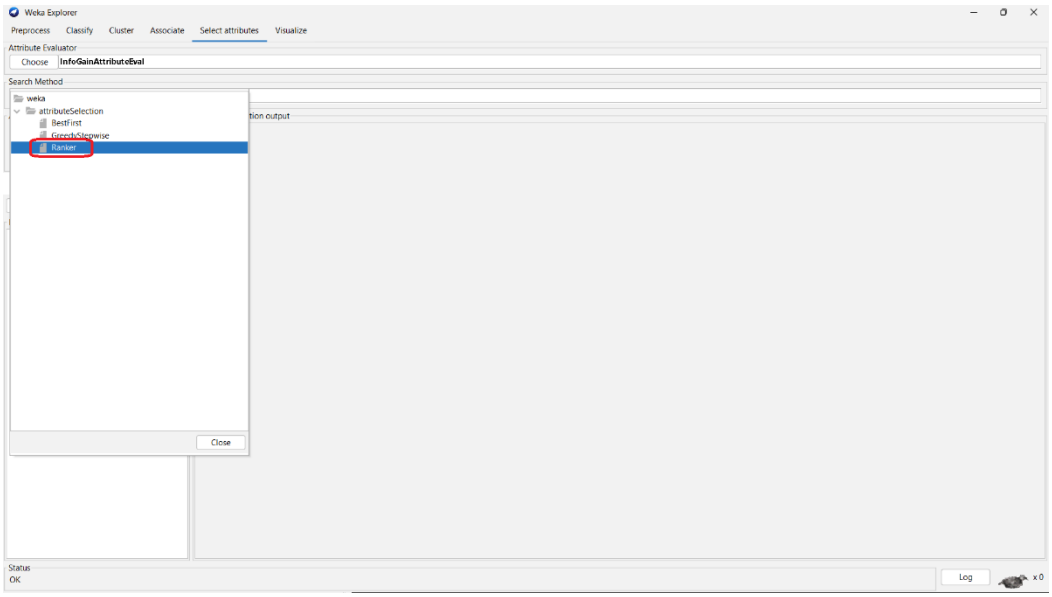
– Which options should be used to select the 5 attributes with the highest correlation? (Step-by-step description, with step-by-step photos and final results)

Trong **Attribute Evaluator** sẽ chọn option **CorrelationAttributeEval**, trong **Search Method** sẽ chọn option **Ranker**

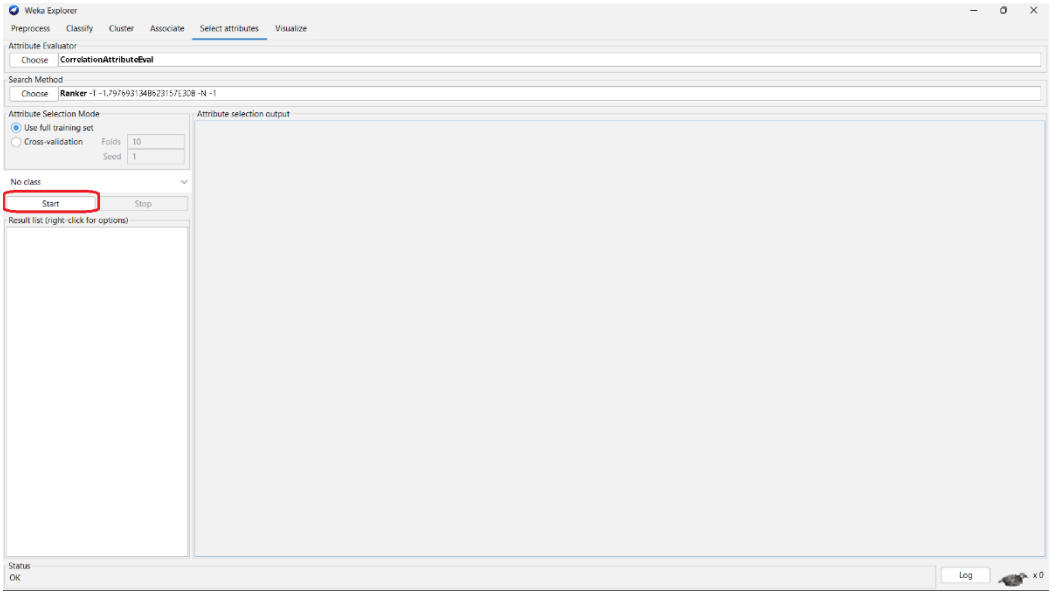
1. Trong **Attribute Evaluator** sẽ chọn option **CorrelationAttributeEval**



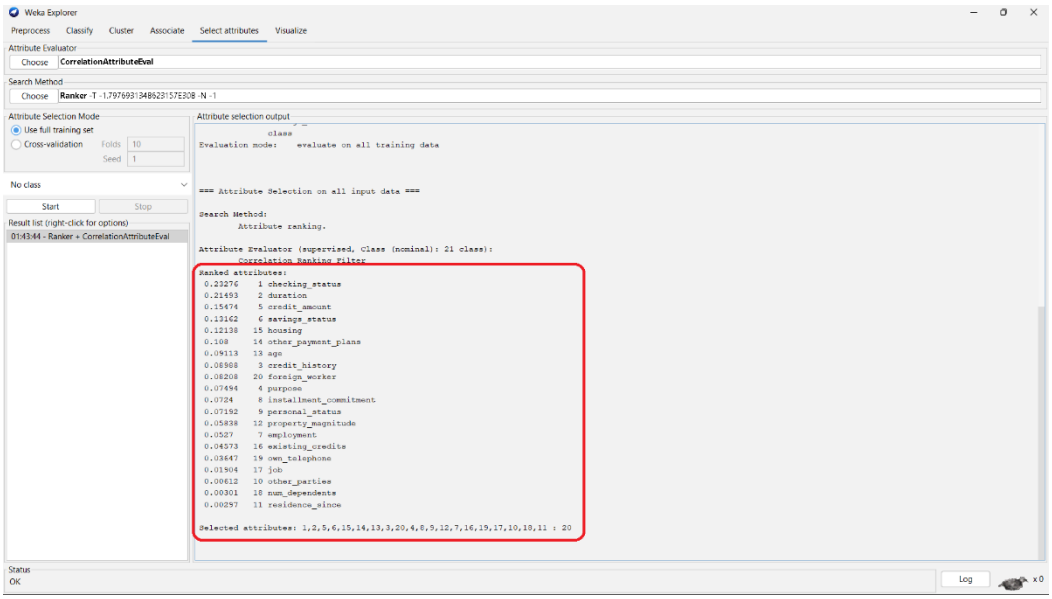
2. Trong **Search Method** sẽ chọn option **Ranker**



3. Chọn **Start**



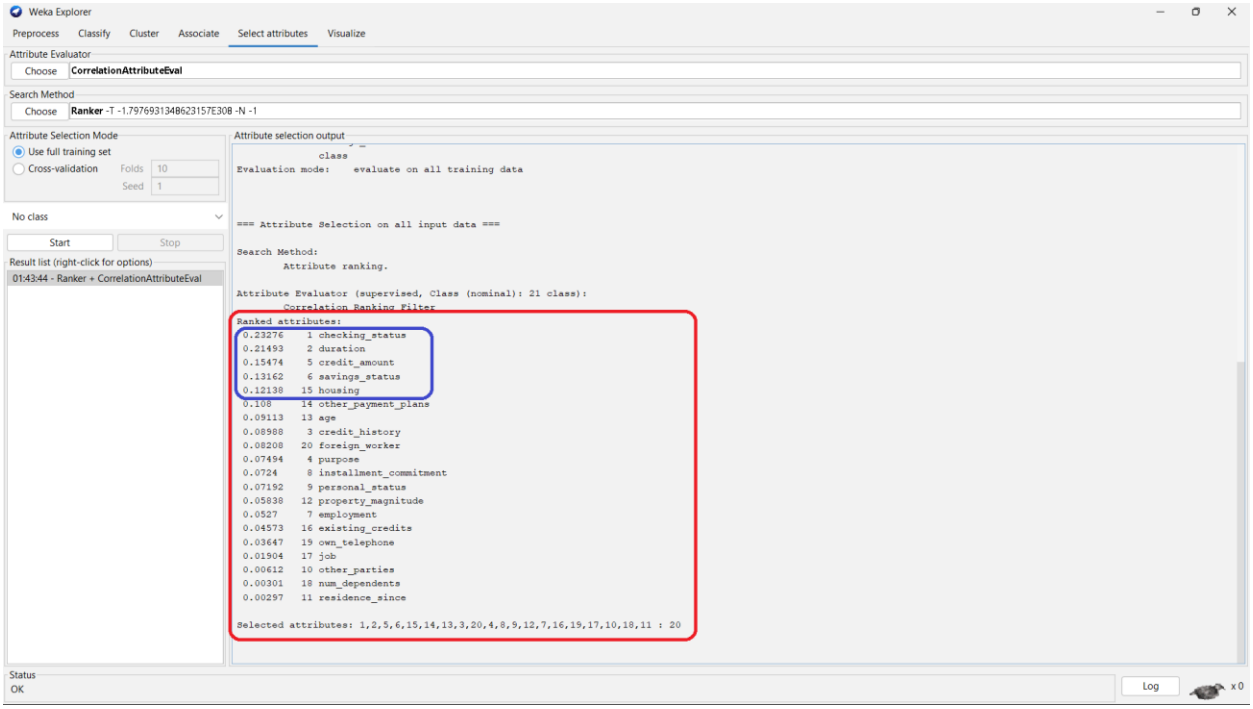
4. Kết quả



Dựa vào kết quả trên màn hình ta có thể thấy được 5 thuộc tính có độ tương quan cao nhất là:

- Checking_status
- Duration

- Credit_amount
- Savings_status
- Housing



3. Preprocessing Data in Python

Câu 1: Trích xuất các cột bị thiếu giá trị

Các bước thực hiện như sau:

- **Bước 1:** Đọc dữ liệu từ tệp và lưu vào mảng dữ liệu
- **Bước 2:** Tìm chỉ số của các cột bị thiếu giá trị
- **Bước 3:** Lưu tên các cột bị thiếu giá trị và ghi nội dung vào tệp Q1.txt

Mã nguồn:

```python
import csv
import io

# Đọc dữ liệu từ file và lưu vào một mảng dữ liệu
data = []
with open("house-prices.csv", "r") as f:
    for line in f:
        data.append(line.strip().split(","))

# Tìm chỉ số của các cột bị thiếu giá trị
missing_columns = []
for i in range(len(data[0])):
    # Kiểm tra xem cột thứ i có chứa giá trị thiếu không
    if any([row[i] == "" for row in data]):
        missing_columns.append(i)

# Lưu tên các cột bị thiếu giá trị vào file txt
column_names = data[0]
missing_column_names = [column_names[i] for i in missing_columns]
with io.open("Q1.txt", mode="w", encoding="utf-8") as f:
    f.write("\n".join(missing_column_names))

```

Câu 2: Đếm số dòng bị thiếu dữ liệu

Các bước thực hiện như sau:

- **Bước 1:** Đọc dữ liệu từ tệp và lưu vào mảng dữ liệu
- **Bước 2:** Duyệt qua từng dòng dữ liệu và tăng biến đếm nếu tồn tại giá trị rỗng
- **Bước 3:** Lưu số dòng bị thiếu dữ liệu vào tệp Q2.txt

Mã nguồn:

```python
import csv
import io

# Đọc dữ liệu từ file và lưu vào một mảng
data = []
with open("house-prices.csv", "r") as f:
    for line in f:
        data.append(line.strip().split(","))

# Khai báo biến đếm số dòng bị thiếu dữ liệu và duyệt qua từng dòng
missing_row_count = 0
for row in data:
    if any([cell == "" for cell in row]):
        missing_row_count += 1

# Ghi số dòng bị thiếu dữ liệu vào file text
with io.open("Q2.txt", mode="w", encoding="utf-8") as f:
    f.write(str(missing_row_count))

```

Câu 3: Điền vào giá trị còn thiếu bằng cách sử dụng giá trị trung bình, trung bình (đối với thuộc tính số) và chế độ (đối với thuộc tính danh mục).

Câu 3.1 Điền vào giá trị còn thiếu bằng cách sử dụng mean

Các bước thực hiện:

- **Bước 1:** Đọc dữ liệu từ tệp và lưu vào mảng dữ liệu
- **Bước 2:** Tính giá trị trung bình mean của thuộc tính số
- **Bước 3:** Lặp lại từng dòng của danh sách dữ liệu và điền giá trị còn thiếu bằng giá trị trung bình mean (nếu có)
- **Bước 4:** Ghi dữ liệu đã được cập nhật vào Q3-mean.csv

Mã nguồn:

```python
import csv

# Đọc tệp csv và lưu dữ liệu vào danh sách
data = []
with open('house-prices.csv') as f:
    reader = csv.reader(f)
    for row in reader:
        data.append(row)

# Tính giá trị trung bình của thuộc tính số
mean_val = {}
for i in range(len(data[0])):
    try:
        sum_val = 0
        cnt_val = 0
        for j in range(1, len(data)):
            if data[j][i] != '':
                sum_val += float(data[j][i])
                cnt_val += 1
        if cnt_val > 0:
            mean_val[i] = sum_val / cnt_val
        else:
            sum_all = 0
            cnt_all = 0
            for j in range(1, len(data)):
                for k in range(len(data[j])):
                    if k not in mean_val and data[j][k] != '':
                        sum_all += float(data[j][k])
                        cnt_all += 1
            overall_mean = sum_all / cnt_all
            mean_val[i] = overall_mean
    except ValueError:
        continue

# Điền giá trị còn thiếu bằng giá trị trung bình
for i in range(1, len(data)):
    for j in range(len(data[i])):
        if data[i][j] == '':
            if j in mean_val:
                data[i][j] = mean_val[j]



# Ghi danh sách dữ liệu đã được cập nhật vào tệp csv
with open('Q3-mean.csv', mode='w', newline='') as f:
    writer = csv.writer(f)
    writer.writerows(data)

```

Câu 3.2 Điền vào giá trị còn thiếu bằng cách sử dụng giá trị median

Các bước thực hiện:

- **Bước 1:** Ta viết hàm tính giá trị median
- **Bước 2:** Đọc dữ liệu từ tệp và lưu vào mảng dữ liệu
- **Bước 3:** Tính giá trị median của tất cả các thuộc tính số

- **Bước 4:** Tính giá trị median của danh sách các giá trị số
- **Bước 5:** Điền giá trị còn thiếu bằng giá trị median (nếu có)
- **Bước 6:** Ghi Ghi dữ liệu đã được cập nhật vào Q3-median.csv

Mã nguồn:

```python
import csv

# Hàm tính giá trị trung vị
def median(lst):
    n = len(lst)
    s = sorted(lst)
    if n % 2 == 0:
        # Nếu số lượng phần tử là số chẵn, chọn giá trị ở giữa
        return (s[n//2-1] + s[n//2]) / 2
    else:
        # Nếu số lượng phần tử là số lẻ, chọn giá trị ở giữa
        return s[n//2]

# Đọc tệp csv và lưu dữ liệu vào danh sách
data = []
with open('house-prices.csv') as f:
    reader = csv.reader(f)
    for row in reader:
        data.append(row)

# Tính giá trị trung vị của thuộc tính số
median_val = {}
for i in range(len(data[0])):
    try:
        val_list = []
        for j in range(1, len(data)):
            if data[j][i] != '':
                val_list.append(float(data[j][i]))
        if val_list:
            median_val[i] = median(val_list)
    except ValueError:
        continue

```

```
33
34  # Tính giá trị trung vị của tất cả các thuộc tính số
35  all_val_list = []
36  for i in range(1, len(data)):
37      for j in range(len(data[i])):
38          try:
39              val = float(data[i][j])
40              all_val_list.append(val)
41          except ValueError:
42              # Nếu giá trị không phải số, ta bỏ qua hoặc thay bằng giá trị khác tùy thích
43              all_val_list.append(0)  # Thay bằng giá trị 0
44              # continue  # Bỏ qua giá trị không phải số
45
46  # Tính giá trị trung vị của danh sách các giá trị số
47  overall_median = median(all_val_list)
48
49
50  # Điền giá trị còn thiếu bằng giá trị trung vị
51  for i in range(1, len(data)):
52      for j in range(len(data[i])):
53          if data[i][j] == '':
54              if j in median_val:
55                  data[i][j] = median_val[j]
56
57  # Ghi danh sách dữ liệu đã được cập nhật vào tệp csv
58  with open('Q3-median.csv', mode='w', newline='') as f:
59      writer = csv.writer(f)
60      writer.writerows(data)
61
```

Câu 3.3 Điền vào giá trị còn thiếu bằng cách sử dụng giá trị mode

Các bước thực hiện:

- **Bước 1:** Đọc dữ liệu từ tệp và lưu vào mảng dữ liệu
- **Bước 2:** Tìm các thuộc tính bị thiếu giá trị. Đối với mỗi thuộc tính thiếu giá trị, ta tạo một danh sách các giá trị xuất hiện trong thuộc tính đó
- **Bước 3:** Chế độ tìm giá trị của danh sách giá trị thuộc tính, nghĩa là giá trị xuất hiện nhiều nhất trong danh sách
- **Bước 4:** Điền giá trị chế độ vào các ô trống (có giá trị trống hoặc giá trị bằng NULL) trong tệp CSV
- **Bước 5:** Lưu kết quả vào tệp Q3-mode.csv

Mã nguồn:

```python
import csv

filename_input = "house-prices.csv"
filename_output = "Q3-mode.csv"
delimiter = ","
missing_values = ["", "NULL", "NaN"] # Giá trị bị thiếu cần điền

# Bước 1: Đọc dữ liệu từ tệp CSV vào danh sách list
data = []
with open(filename_input, "r") as file:
    for row in file:
        fields = row.strip().split(delimiter)
        data.append(fields)

# Bước 2: Tìm các thuộc tính thiếu giá trị
num_cols = len(data[0])
missing_cols = []
for i in range(num_cols):
    values = [row[i] for row in data if row[i] not in missing_values]
    if len(values) != len(data):
        missing_cols.append((i, values))

# Bước 3: Tìm giá trị mode của các thuộc tính thiếu giá trị
modes = []
for col, values in missing_cols:
    freqs = {value: values.count(value) for value in set(values)}
    mode_value = max(freqs, key=freqs.get)
    modes.append((col, mode_value))

# Bước 4: Điền giá trị mode vào các ô Trống
for row in data:
    for col, mode_value in modes:
        if row[col] in missing_values:
            row[col] = mode_value

# Bước 5: Lưu kết quả vào tệp CSV
with open(filename_output, "w") as file:
    for row in data:
        file.write(delimiter.join(row) + "\n")

```

Câu 4: Xóa các hàng chứa nhiều hơn một số giá trị cụ thể bị thiếu

Các bước thực hiện như sau:

- **Bước 1:** Đọc dữ liệu từ tệp csv vào một mảng dữ liệu
- **Bước 2:** Loại bỏ hàng bị thiếu dữ liệu (ở đây ví dụ số lượng giá trị thiếu nhiều hơn 5%)
- **Bước 3:** Ghi dữ liệu mới vào tệp Q4.csv

Mã nguồn:

```python
import csv

delimiter = ','

# Đọc dữ liệu từ tệp csv
with open('house-prices.csv', 'r') as file:
    data = []
    for row in file:
        data.append(row.strip().split(delimiter))

# Loại bỏ hàng bị thiếu dữ liệu
new_data = []
for row in data:
    num_missing = 0
    for value in row:
        if value == '':
            num_missing += 1
    if num_missing <= int(len(row)*0.05):
        new_data.append(row)

# Ghi dữ liệu mới vào tệp csv
with open('Q4.csv', 'w', newline='') as file:
    writer = csv.writer(file, delimiter=delimiter)
    for row in new_data:
        writer.writerow(row)

```

Câu 5: Xóa các cột chứa nhiều hơn một số giá trị cụ thể bị thiếu

Các bước thực hiện như sau:

- **Bước 1:** Đọc dữ liệu từ tệp csv vào một mảng dữ liệu
- **Bước 2:** Dùng vòng lặp để tìm kiếm các cột có số giá trị bị thiếu vượt quá ngưỡng cho phép (ở đây ví dụ số lượng giá trị thiếu nhiều hơn 5%)
- **Bước 3:** Dùng vòng lặp để loại bỏ các cột đó
- **Bước 4:** Ghi dữ liệu mới vào file Q5.csv

Mã nguồn:

```python
import csv

delimiter = ','

# Đọc dữ liệu từ tệp csv
with open('house-prices.csv', 'r') as file:
    data = []
    for row in file:
        data.append(row.strip().split(delimiter))

# Tìm cột bị thiếu dữ liệu
missing_columns = []
num_rows = len(data)
num_cols = len(data[0])
for j in range(num_cols):
    num_missing = 0
    for i in range(num_rows):
        if data[i][j] == '':
            num_missing += 1
    if num_missing > int(num_rows*0.05):
        missing_columns.append(j)

# Loại bỏ cột bị thiếu dữ liệu
new_data = []
for row in data:
    new_row = []
    for j, value in enumerate(row):
        if j not in missing_columns:
            new_row.append(value)
    new_data.append(new_row)

# Ghi dữ liệu mới vào tệp csv
with open('Q5.csv', 'w', newline='') as file:
    writer = csv.writer(file, delimiter=delimiter)
    for row in new_data:
        writer.writerow(row)


```

Câu 6: Xóa các mẫu trùng lặp.

Các bước thực hiện như sau:

- **Bước 1:** Đọc dữ liệu từ tệp csv vào một mảng dữ liệu
- **Bước 2:** Tạo một danh sách mới để lưu trữ các mẫu không trùng lặp
- **Bước 3:** Sử dụng vòng lặp để xác định các mẫu lặp lại bằng cách so sánh từng mẫu với các mẫu khác trong danh sách. Nếu một mẫu đã tồn tại trong danh sách mới, ta sẽ bỏ qua mẫu này và tiếp tục với các mẫu khác. Nếu một mẫu chưa tồn tại trong danh sách mới, ta sẽ bổ sung nó vào danh sách này.

- **Bước 4:** Ghi các mẫu không trùng lặp vào tệp Q6.csv

Mã nguồn:

```python
import csv

delimiter = ','

# Đọc bảng dữ liệu từ tệp csv
with open('house-prices.csv', 'r') as file:
    data = []
    csv_reader = csv.reader(file, delimiter=delimiter)
    for row in csv_reader:
        data.append(row)

# Xác định các mẫu không trùng lặp
new_data = []
for row in data:
    if row not in new_data:
        new_data.append(row)

# Ghi các mẫu không trùng lặp vào một tệp csv mới
with open('Q6.csv', 'w', newline='') as file:
    csv_writer = csv.writer(file, delimiter=delimiter)
    for row in new_data:
        csv_writer.writerow(row)


```

Câu 7: Bình thường hóa một thuộc tính số sử dụng các phương pháp min-max và Z-score.

Câu 7.1 Phương pháp min-max

Các bước thực hiện như sau:

- **Bước 1:** Tạo biến filename để chỉ đến tệp cho trước, biến output_filename để đặt tên cho tệp mới sau khi bình thường hóa xong, và biến index_attribute để chỉ định index cột bạn muốn bình thường hóa.
- **Bước 2:** Đọc tệp và lưu dữ liệu vào danh sách các hàng
- **Bước 3:** Tìm giá trị nhỏ nhất và lớn nhất của cột được chỉ định bằng biến "index_attribute" bằng vòng lặp.

- **Bước 4:** Duyệt qua danh sách các hàng và thực hiện bình thường hóa
- **Bước 5:** Ghi dữ liệu mới đã bình thường hóa vào tệp Q7-minmax.csv

Mã nguồn:

```python
import csv

# đặt tên tệp csv để bình thường hóa và tạo biến "index_attribute" để chỉ định index của cột muốn bình thường hóa
filename = "house-prices.csv"
output_filename = "Q7-minmax.csv"
index_attribute = 0  #ví dụ ở đây là index cột 0 là Id

# đọc tệp csv bằng module csv và lưu dữ liệu vào danh sách các hàng
rows = []
with open(filename, 'r') as csvfile:
    csvreader = csv.reader(csvfile)
    for row in csvreader:
        rows.append(row)

# tìm giá trị nhỏ nhất và lớn nhất của cột được chỉ định bằng biến "index_attribute"
min_value = float('inf')
max_value = -float('inf')
for row in rows:
    try:
        value = float(row[index_attribute])
        if value < min_value:
            min_value = value
        if value > max_value:
            max_value = value
    except ValueError:
        pass

# duyệt qua danh sách các hàng và thực hiện bình thường hóa
for row in rows[1:]:  # bỏ qua hàng đầu tiên chứa tiêu đề
    try:
        value = float(row[index_attribute])
        normalized_value = (value - min_value) / (max_value - min_value)
        row[index_attribute] = normalized_value
    except ValueError:
        pass

# ghi dữ liệu mới đã bình thường hóa và chuyển đổi thành số nguyên vào một tệp csv mới
with open(output_filename, 'w', newline='') as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerows(rows)

```

Câu 7.2 Phương pháp Z-score

Các bước thực hiện như sau:

- **Bước 1:** Tạo biến filename để chỉ đến tệp cho trước, biến output_filename để đặt tên cho tệp mới sau khi bình thường hóa xong, và biến index_attribute để chỉ định index cột bạn muốn bình thường hóa.
- **Bước 2:** Đọc tệp và lưu dữ liệu vào danh sách các hàng

- **Bước 3:** Tính giá trị trung bình (mean) và độ lệch chuẩn (standard deviation) của thuộc tính muốn chuẩn hóa
- **Bước 5:** Duyệt qua danh sách các hàng và thực hiện chuẩn hóa Z-score
- **Bước 6:** Ghi dữ liệu mới đã chuẩn hóa Z-score vào tệp Q7-Zscore.csv

Mã nguồn:

```
import csv

# đặt tên tệp csv để bình thường hóa và tạo biến "index_attribute" để chỉ định index của cột muốn bình thường hóa
filename = "house-prices.csv"
output_filename = "Q7-Zscore.csv"
index_attribute = 0

# đọc tệp csv bằng module csv và lưu dữ liệu vào danh sách các hàng
rows = []
with open(filename, 'r') as csvfile:
    csvreader = csv.reader(csvfile)
    for row in csvreader:
        rows.append(row)

# tạo danh sách các giá trị của thuộc tính muốn chuẩn hóa
data = []
for row in rows[1:]:  # bỏ qua hàng đầu tiên chứa tiêu đề
    try:
        value = float(row[index_attribute])
        data.append(value)
    except ValueError:
        pass


# tính giá trị trung bình (mean) và độ lệch chuẩn (standard deviation) của thuộc tính muốn chuẩn hóa
mean = sum(data) / len(data)
stdev = math.sqrt(sum([(x - mean) ** 2 for x in data]) / (len(data) - 1))


# duyệt qua danh sách các hàng và thực hiện chuẩn hóa Z-score
for row in rows[1:]:  # bỏ qua hàng đầu tiên chứa tiêu đề
    try:
        value = float(row[index_attribute])
        z_score = (value - mean) / stdev
        row[index_attribute] = z_score
    except ValueError:
        pass

# ghi dữ liệu mới đã chuẩn hóa Z-score vào một tệp csv mới
with open(output_filename, 'w', newline='') as csvfile:
    csvwriter = csv.writer(csvfile)
    csvwriter.writerows(rows)

```

Câu 8: Thực hiện cộng, trừ, nhân, chia hai thuộc tính số.

Các bước thực hiện như sau:

- **Bước 1:** Đọc tệp và lưu dữ liệu vào danh sách
- **Bước 2:** Thêm các tiêu đề (Add, Sub, Mul, Div)
- **Bước 3:** Dùng vòng lặp duyệt để lấp đầy ô trống (trống ta coi như nó bằng 0). Vòng lặp này sinh ra để tránh lỗi tính toán với ô trống

- **Bước 4:** Tính toán các kết quả cho vào danh sách dữ liệu (có sử dụng hàm isnumeric() để kiểm xem giá trị có phải số không)
- **Bước 5:** Ghi kết quả vào tệp Q8.csv

Mã nguồn:

```python
import csv

# Ví dụ ta thử tính toán với cột 0 và cột 1 của tệp
col1 =0
col2 =1
# đọc dữ liệu từ tệp csv vào list
data = []
with open('house-prices.csv', 'r') as csvfile:
    reader = csv.reader(csvfile)
    for row in reader:
        data.append(row)

# thêm các tiêu đề cho các cột mới
data[0].append('Add')
data[0].append('Sub')
data[0].append('Mul')
data[0].append('Div')

# dùng vòng lặp duyệt để lấp đầy ô trống (trống ta coi như nó bằng 0)
# vòng lặp này sinh ra để tránh lỗi tính toán với ô trống
for i in range(1, len(data)):
    if data[i][col1]=="":
        (data[i][col1])=0;
    if data[i][col2]=="":
        (data[i][col2])=0;

# tính toán và thêm kết quả vào list data
for i in range(1, len(data)):
    number1 = float(data[i][col1])
    number2 = float(data[i][col2])

    add = str(number1 + number2)
    sub = str(number1 - number2)
    mul = str(number1 * number2)
    if number2 != 0:
        div = str(number1 / number2)
    else:
        div = 'NaN'
    data[i].append(add)
    data[i].append(sub)
    data[i].append(mul)
    data[i].append(div)

# ghi kết quả vào file mới
with open('Q8.csv', 'w', newline='') as csvfile:
    writer = csv.writer(csvfile, delimiter=',')
    for row in data:
        writer.writerow(row)

```

III. TÀI LIỆU THAM KHẢO

1. https://web888.vn/tien-xu-ly-du-lieu-trong-machine-learning-vi-du-cu-the/
2. https://t3h.edu.vn/tin-tuc/tep-csv-trong-python-lam-viec-voi-tep-csv-trong-python#:~:text=reader%20(),li%E1%BB%87u%20CSV%20%C4%91%C6%B0%E1%BB%A3c%20ch%E1%BB%89%20%C4%91%E1%BB%8Bnh.
3. https://yeulaptrinh.vn/doc-ghi-file-csv-trong-python/
4. https://www.geeksforgeeks.org/data-preprocessing-machine-learning-python/

5. https://monashdatafluency.github.io/python-workshop-base/modules/missing_values/
6. https://t3h.edu.vn/tin-tuc/tep-csv-trong-python-lam-viec-voi-tep-csv-trong-python
7. https://www.codecademy.com/article/normalization