HƯỚNG DẪN CHẠY CHƯƠNG TRÌNH

☆ Chương trình sử dụng tham số dòng lệnh:

`python source.py <tên file csv> <mã số của chức năng>`

`<tên file csv>` : chương trình sẽ lấy dữ liệu từ file csv này để chạy

`<mã số của chức năng>` : là 1 số nguyên trong khoảng [1…8]

⁂ Tùy từng chức năng sẽ yêu cầu thêm những tham số phụ.

☆ Chi tiết về tham số của từng chức năng

1. Liệt kê các cột dữ liệu bị thiếu

`python source.py <tên file csv> 1`

Ví dụ:

`python source.py temperature.csv 1`

Chương trình đọc dữ liệu từ file *temperature.csv* và liệt kê các cột dữ liệu bị thiếu ra màn hình console.

2. Đếm số dòng bị thiếu dữ liệu

`python source.py <tên file csv> 2`

Ví dụ:

`python source.py temperature.csv 2`

Chương trình đọc dữ liệu từ file *temperature.csv* và in số dòng bị thiếu dữ liệu ra màn hình console.

3. Điền giá trị bị thiếu bằng phương pháp mean, median (cho thuộc tính numeric) và mode (cho thuộc tính categorical)

`python source.py <tên file csv> 3 -m <phương pháp> -attr <tên cột> -o <tên file in kết quả>`

`<phương pháp>` : chỉ nhận các giá trị *mean*, *median* hoặc *mode*; có thể 1 hoặc nhiều giá trị, các giá trị cách nhau bởi khoảng trắng

`<tên cột>` : cột muốn điền giá trị bị thiếu; có thể 1 hoặc nhiều giá trị, các giá trị cách nhau bởi khoảng trắng; tên cột phải đúng với tên cột có trong file dữ liệu vào (phân biệt hoa – thường)

- Nếu truyền vào 1 `<phương pháp>` và nhiều `<tên cột>`, các cột sẽ sử dụng cùng 1 phương pháp được truyền.
- Nếu truyền vào nhiều `<phương pháp>` và nhiều `<tên cột>`, số lượng `<phương pháp>` và `<tên cột>` phải bằng nhau. Mỗi cột sẽ sử dụng phương pháp tương ứng theo thứ tự được truyền.
- Dữ liệu của cột tính mean hoặc median phải đảm bảo thuộc kiểu numeric.

`<tên file in kết quả>` : là file csv dùng để lưu kết quả

Ví dụ:

```
python source.py score.csv 3 -m mean -attr math Cs physic -o filled_score.csv
```

Chương trình dọc dữ liệu từ file *score.csv*; điền dữ liệu bị thiếu vào các cột math, Cs, physic bằng phương pháp mean; và lưu kết quả vào file *filled_score.csv*

```
python source.py score.csv 3 -m mean median mean mode -attr math physic AI Result -o filled_score.csv
```

Chương trình đọc dữ liệu từ file *score.csv*; điền dữ liệu bị thiếu vào các cột bằng phương pháp tương ứng: math – mean, physic – median, AI – mean, Result – mode; lưu kết quả vào file *filled_score.csv*

4. Xóa các dòng bị thiếu dữ liệu với ngưỡng tỉ lệ thiếu cho trước

```
python source.py <tên file csv> 4 -ratio <tỉ lệ> -o <tên file in kết quả>
```

`<tỉ lệ>` : là 1 số thực trong khoảng [0…100]; là ngưỡng tỉ lệ để so sánh

`<tên file in kết quả>` : là file csv dùng để lưu kết quả

Ví dụ:

```
python source.py goods.csv 4 -ratio 77 -o result.csv
```

Chương trình đọc dữ liệu từ file *goods.csv*; xóa những dòng bị thiếu hơn 77% giá trị các thuộc tính; lưu kết quả vào file *result.csv*

5. Xóa các cột bị thiếu dữ liệu với ngưỡng tỉ lệ thiếu cho trước

```
python source.py <tên file csv> 5 -ratio <tỉ lệ> -o <tên file in kết quả>
```

`<tỉ lệ>` : là 1 số thực trong khoảng [0…100]; là ngưỡng tỉ lệ để so sánh

`<tên file in kết quả>` : là file csv dùng để lưu kết quả

Ví dụ:

```
python source.py goods.csv 5 -ratio 83.5 -o result.csv
```

Chương trình đọc dữ liệu từ file *goods.csv*; xóa những cột bị thiếu giá trị thuộc tính ở hơn 83.5% số mẫu; lưu kết quả vào file *result.csv*

6. Xóa các mẫu bị trùng lặp

```
python source.py <tên file csv> 6 -o <tên file in kết quả>
```

`<tên file in kết quả>` : là file csv dùng để lưu kết quả

☞ File dữ liệu đầu vào cần đảm bảo cột đầu tiên là mã phân biệt giữa các mẫu.

Ví dụ:

```
python source.py opinions.csv  6 -o result.csv
```

Chương trình đọc dữ liệu từ file *opinions.csv*; xóa các mẫu bị trùng lặp; lưu dữ liệu vào file *result.csv*

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

```
python source.py <tên file csv> 7 -m <phương pháp> -attr <tên cột> -o <tên file in kết quả>
```

`<phương pháp>` : chỉ nhận giá trị *minmax* hoặc *zscore*; có thể 1 hoặc nhiều giá trị, các giá trị cách nhau bởi khoảng trắng

`<tên cột>` : tên cột muốn chuẩn hóa; có thể 1 hoặc nhiều giá trị, các giá trị cách nhau bởi khoảng trắng; tên cột phải đúng với tên cột có trong file dữ liệu vào (phân biệt hoa – thường); dữ liệu của cột phải đảm bảo thuộc kiểu numeric

- Nếu truyền vào 1 `<phương pháp>` và nhiều `<tên cột>`, các cột sẽ sử dụng cùng 1 phương pháp được truyền.
- Nếu truyền vào nhiều `<phương pháp>` và nhiều `<tên cột>`, số lượng `<phương pháp>` và `<tên cột>` phải bằng nhau. Mỗi cột sẽ sử dụng phương pháp tương ứng theo thứ tự được truyền.
- Chuẩn hóa min-max được mặc định là chuẩn hóa về khoảng [0, 1]

`<tên file in kết quả>` : là file csv dùng để lưu kết quả

Ví dụ:

```
python source.py city.csv 7 -m minmax -attr pop Area -o normalize.csv
```

Chương trình đọc dữ liệu từ file *city.csv*; chuẩn hóa thuộc tính pop và Area bằng phương pháp min-max; lưu kết quả vào file *normalize.csv*

```
python source.py city.csv 7 -m zscore minmax zscore -attr pop Area GDP -o normalize.csv
```

Chương trình đọc dữ liệu từ file *city.csv*; chuẩn hóa các thuộc tính bằng phương pháp tương ứng: pop – Z-score, Area – min-max, GDP – Z-score; lưu kết quả vào file *normalize.csv*

8. Tính giá trị biểu thức thuộc tính

```
python source.py <tên file csv> 8 -exp <biểu thức> -nc <tên cột mới> -o <tên file in kết quả>
```

`<biểu thức>` : biểu thức tính toán giữa các thuộc tính, viết liền không khoảng trắng; các thuộc tính trong biểu thức phải đảm bảo có trong file dữ liệu vào (phân biệt hoa – thường)

`<tên cột mới>` : tên của cột lưu kết quả tính biểu thức

`<tên file in kết quả>` : là file csv dùng để lưu kết quả

Ví dụ:

```
python source.py house-prices.csv 8 -exp 1stFlrSF+2ndFlrSF -nc newCol -o newfile.csv
```

Chương trình đọc dữ liệu từ file *house-prices.csv*; tạo 1 cột mới tên là newCol với các giá trị được tính bằng biểu thức 1stFlrSF + 2ndFlrSF; lưu kết quả vào file *newfile.csv*