Đây là tài liệu đóng vai trò hướng dẫn sử dụng phần mềm Đây là bản copy (`.pdf`) của tài liệu gốc (`.md`) trong thư mục `docs/` , được viết để phục vụ đồ án, có thể sẽ không được cập nhật trong tương lai.

Lời tựa

Phần mềm preprocessor được viết bằng ngôn ngữ python, sử dụng python3 và sử dụng CLI để tương tác.

Cấu trúc thư mục

```
.
├── data
│   └── house-prices.csv
├── docs
│   ├── requirements.txt
│   └── usage.md
├── src
    ├── lib
    │   ├── __init__.py
    │   ├── preprocessor.py
    │   ├── stack.py
    │   └── xfix.py
    └── preprocess.py


4 directories, 9 files
```

Dependency

Ngoài các thư viện có sẵn của python, phần mềm còn sử dụng thêm pakage `tabulate` để hỗ trợ hiển thị đồ họa, ta có thể cài đặt đơn giản:

```
pip install tabulate
```

Hoặc có thể được cài đặt qua file `requirements.txt` trong thư mục `docs/`

```
pip install - r requirements.txt
```

Usage

Chúng ta có thể tương tác với phần mềm qua CLI, với script đóng vai trò tương tác chính là `preprocess.py` trong thư mục `src/` Những câu lệnh được nói đến ở đây chỉ bao gồm những lệnh cần thiết cho quá trình thực hiện bài tập `3` của đồ án

help

Để xem các thông tin liên quan về cách sử dụng, ta có thể chạy lệnh sau

```
python3 preprocess.py -h
```

Console sẽ liệt kê các hàm hỗ trợ cũng như cách thức sử dụng các hàm này. Để xem thêm thông tin về cách sử dụng của từng hàm, ta có thể dùng lệnh

```
python3 preprocess.py <option> -h
```

Với `option` là tên hàm, ví dụ `python3 preprocess.py fill -h` để xem thông tin sử dụng hàm điền giá trị null và các flag liên quan Với từng câu lệnh trên, console cũng sẽ hiển thị các `flag` (`-flag`) có thể sử dụng và ý nghĩa liên quan, do hướng dẫn này đã khá chi tiết nên sẽ không nói lại chi tiết ở đây, file này chỉ tập trung vào cách đạt được các yêu cầu trong đồ án.

Tuy nhiên cần lưu ý tới flag `-f` , `--file` tượng trưng cho file đầu vào để xử lý, giá trị của flag này trong xuyên suốt file hướng dẫn sẽ được để cố định trỏ tới `../data/house-prices.csv` , trong thực tế sử dụng, trường này có thể là bất kì giá trị nào trên ổ cứng trỏ tới file dữ liệu

Một số hàm cho phép gán biến file đầu ra `-o` , `--outfile` , flag này không bắt buộc, tuy nhiên nếu không cung cấp giá trị biến đầu ra nào, kết quả dữ liệu sau khi chạy hàm sẽ được lưu đè lên file đầu vào ở `-f` . Đây cũng là một điểm quan trọng cần lưu ý để tránh mất hoặc thay đổi dữ liệu, để bảo tồn file gốc ở đây sẽ luôn sử dụng flag này.

Liệt kê thông tin về các dòng, cột bị thiếu dữ liệu

```
 python3 preprocess.py list -h
```

Ở đây cần chú ý 3 flag chính đó là:

```
 -mr, --missing-rows     flag yêu cầu thông tin về dòng
-mc, --missing-cols     flag yêu cầu thông tin về cột
-m, --missing           flag yêu cầu thông tin về cả dòng và cột
```

Liệt kê các cột bị thiếu dữ liệu

```
python3 preprocess.py -f ../data/house-prices.csv list -mc
```

Kết quả đầu ra sẽ là một bảng sử dụng thư viện `tabulate` để vẽ

Đếm số dòng bị thiếu dữ liệu

```
python3 preprocess.py -f ../data/house-prices.csv list -mr
```

Điền giá trị thiếu

```
python3 preprocess.py fill -h
```

Ở đây có 1 số flag quan trọng

```
 -ft, --filltype     biến lựa chọn kiểu điền dữ liệu numeric, có thể là mean hoặc median
-fb, --fallback     trong trường hợp quá trình điền dữ liệu thất bại, giá trị ở đây sẽ làm giá trị thay thế
-o, --outfile       file đầu ra
```

Giá trị của `-ft` chỉ ảnh hưởng đến các thuộc tính NUMERIC, các thuộc tính thuộc dạng CATEGORICAL luôn sử dụng phương pháp mode là điền thiếu

Điền bằng phương pháp mean

```
python3 preprocess.py -f ../data/house-prices.csv fill -ft mean -o mean_fill.csv
```

`mean_fill.csv` là file output chứa kết quả

Điền bằng phương pháp median

```
python3 preprocess.py -f ../data/house-prices.csv fill -ft median -o median_fill.csv
```

`median_fill.csv` là file output chứa kết quả

Xóa các dòng, cột bị thiếu dữ liệu

```
python3 preprocess.py delthres -h
```

Một số flag cần lưu ý

```
 -t, --type              biến lựa chọn, có thể là row hoặc col, thể hiện loại cần xóa
-ti, --threshold-int    định nghĩa cụ thể giá trị mức, dữ liệu thiếu bằnghoặc vượt mức này sẽ bị xóa
-tp , --threshold-percentage  định nghĩa giá trị mức nhưng theo phần trăm số dòng hoặc cột, nằm trong khoảng từ 0-1, nếu điền giá tr
-o, --outfile       file đầu ra
```

Xóa các dòng bị thiếu với ngưỡng cho trước

Xóa những dòng thiếu 20 giá trị thuộc tính và lưu kết quả vào `del_row_int.csv`

```
python3 preprocess.py -f ../data/house-prices.csv delthres -t row -ti 20 -o del_row_int.csv
```

Xóa những dòng thiếu từ 80% giá trị thuộc tính và lưu kết quả vào `del_row_pct.csv`

```
python3 preprocess.py -f ../data/house-prices.csv delthres -t row -tp 0.8 -o del_row_pct.csv
```

Xóa các cột dữ liệu với ngưỡng cho trước

Xóa những cột thiếu 500 giá trị dòng và lưu kết quả vào `del_col_int.csv`

```
python3 preprocess.py -f ../data/house-prices.csv delthres -t col -ti 500 -o del_col_int.csv
```

Xóa những cột thiếu từ 80% giá trị dòng và lưu kết quả vào `del_col_pct.csv`

```
python3 preprocess.py -f ../data/house-prices.csv delthres -t col -tp 0.8 -o del_col_pct.csv
```

Xóa các dòng dữ liệu bị trùng

```
python3 preprocess.py deldup -h
```

```
-t, --type  loại dữ liệu để xóa, hiện tại chỉ hỗ trợ xóa dòng nên chỉ có 1 giá trị row
-o, --outfile file xuất ra
```

Xóa các dòng dữ liệu bị trùng và xuất ra deldup_file.csv

```
python3 preprocess.py -f ../data/house-prices.csv deldup -t row -o deldup_file.csv
```

Chuẩn hóa thuộc tính

```
python3 preprocess.py norm -h
```

```
-t , --type        loại chuẩn hóa, chọn 1 trong ['min-max', 'z-score']
-a , --attribute   tên thuộc tính NUMERIC cần chuẩn hóa
-o , --outfile     file đầu ra
```

Chuẩn hóa thuộc tính theo min-max

Chuẩn hóa thuộc tính LotFrontage và lưu vào norm_file.csv

```
python preprocess.py -f ../data/house-prices.csv norm -t min-max -a LotFrontage -o norm_file.csv
```

Chuẩn hóa thuộc tính theo z-score

Chuẩn hóa thuộc tính LotFrontage và lưu vào zscore_file.csv

```
python preprocess.py -f ../data/house-prices.csv norm -t z-score -a LotFrontage -o zscore_file.csv
```

Tính toán thuộc tính

Hiện tại chỉ hỗ trợ phép +, - , * , /

```
python3 preprocess.py -f ../data/house-prices.csv acalc -h
```

```
 -c , --calc-string phép tính
-a , --attribute-name tên của thuộc tính mới để lưu kết quả
-o , --outfile      tên file đầu ra
```

Phép tính hỗ trợ tính biểu thức toán nhiều thuộc tính, nhiều phép tính, miễn là tất cả các thuộc tính phải tồn tại, đều là thuộc tính NUMERIC và các phép tính chỉ bao gồm +, - , * , /

Ví dụ 1 biểu thức tính toán

```
 python3 preprocess.py -f ../data/house-prices.csv acalc -c '(LotFrontage - OverallQual) * OverallCond' -a test -o newattr.csv
```