Tiền xử lý dữ liệu trong Data Mining

Phan Anh Phong, PhD.

Vinh University

Mục lục

Bài 1: Xử lý dữ liệu bị thiếu giá trị

Cho dataset trong Bảng 1:

Bảng 1

a	b	x	y
1	6	11	16
2	7	12	Null
3	8	Null	Null
4	Null	Null	Null
Null	Null	Null	Null

Giải thích: a, b, x và y là các thuộc tính; Null - thiếu giá trị

Trong Python sử dụng numpy để tạo lập bảng trên, cụ thể như sau:

```
import numpy as np
df = pd.DataFrame({'a':[1,2,3,4,np.nan],
                   'b':[6,7,8,np.nan,np.nan],
                   'x':[11,12,13,np.nan,np.nan],
                   'y':[16,np.nan,np.nan,19,np.nan]})
```

Thực hiện các yêu cầu dưới đây (sử dụng hàm **fillna** trong pandas để thay thế giá trị Null; hàm **dropna** để xóa dòng)

1. Thay thế tất cả giá trị null trong dataset bởi 0
2. Thay thế giá trị null ở mỗi thuộc tính bởi trung vị (median) của các giá trị có trong thuộc tính đó
3. Thay thế giá trị null ở mỗi thuộc tính bởi giá trị mode của các giá trị có trong thuộc tính đó
4. Xóa các dòng có giá trị null
5. Tìm hiểu phương pháp thay thế null với bfill, ffill của fillna và áp dụng vào data set đã cho
6. Tìm hiểu phương thức DataFrame.Describle() trong Pandas và thực hiện, giải thích kết quả thu được (có thể tham khảo ở đây https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html)

7. Nghiên cứu thêm về missing values trong python ở liên kết này https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html. Viết báo cáo nạp qua hệ thống LMS

Lưu ý: Hiển thị nội dung data set trước và sau khi thực hiện các công việc để kiểm tra kết quả thực hiện thuật toán

Bài 2: Xử lý giá trị ngoại lai

1. Phương pháp IQR

Cho tập dữ liệu data gồm các phần tử [12, 13, 12, 10, 10, 11, 12, 15, 12, 16]
Viết chương trình Python thực hiện các công việc dưới đây (sử dụng các hàm sorted, percentitle thư viện numpy và pandas và các cấu trúc lập trình for, if…)

1. Đưa ra màn hình giá trị q1, q2, q3
2. Tính khoảng IQR và đưa kết quả ra màn hình

2. Liệt kê các phần tử ngoại lai
3. Cho biết các phần tử của tập dữ liệu sau khi đã loại bỏ phần tử ngoại lai

2. Phương pháp Zscore

1. Chạy chương trình sau, giải thích ý nghĩa từng lệnh:

```
import numpy as np
import pandas as pd
dataset= [10,12,12,13,12,11,14,13,15,10,10, 12, 100]
outliers = []
def detect_outlier(data_1):
    threshold = 3
    mean_1 = np.mean(data_1)
    std_1 = np.std(data_1)
    for y in data_1:
        z_score = (y - mean_1) / std_1
        if np.abs(z_score) > threshold:
            outliers.append(y)
    return outliers
outlier_datapoints = detect_outlier(dataset)
print('data set: ', dataset)
print('outlier_datapoints of data set: ', outlier_datapoints)
```

2. Đưa ra tập dữ liệu sau khi xóa phần tử ngoại lai

3. Viết báo cáo về phương pháp Zscore khi thực hiện loại phần tử ngoại lai và nạp vào hệ thống LMS

3. Phương pháp dựa vào trung vị

Cho bảng dữ liệu như sau:

Bảng 2.

Object	Income	Age	Class
1	3.000.000	23	Y
2	9.600.000	56	N
3	4.700.000	43	Y
4	7.000.000	30	N
5	6.200.000	65	N
6	2.200.000	26	Y
7	6.600.000	38	N
8	2.000.000	31	Y
9	6.300.000	37	Y
10	7.000.000	42	N
11	8.000.000	47	N
12	100.000.000	51	Y

Viết chương trình python loại dữ liệu ngoại lai bằng phương pháp dựa vào trung vị (tham khảo bài giảng chương 2)

Bài 3. Giảm chiều dữ liệu

Cho data set như Bảng 1, viết chương trình cho biết quan hệ giữa Income và Age bằng phương pháp phân tích tương quan. Nạp kết quả vào hệ thống LMS

Bài 4. Rời rạc hóa dữ liệu

Dữ liệu liên tục thường được rời rạc hóa thành các thùng (bin) để phân tích. Giả sử ta có dữ liệu về tuổi một nhóm người (ages) và muốn rời rạc hóa tuổi của họ vào các nhóm. Hàm **cut** của thư viện **pandas** trong Python sử dụng để thực hiện việc này.

Hãy lần lược thực hiện các yêu cầu sau:

1. Chia tập dữ liệu ages thành các thùng từ 18 đến 25, 26 đến 35, 36 đến 60 và cuối cùng là 61 tuổi trở lên. Chương trình dưới đây sẽ thực hiện yêu cầu trên

```
import pandas as pd
ages = [20, 22, 25, 27, 21, 23, 37, 31, 61, 45, 41, 32]
bins = [18, 25, 35, 60, 100]
cats = pd.cut(ages, bins)
print(cats)
print(cats.codes)
print(cats.categories)
print(pd.value_counts(cats))
```

Chạy chương trình và giải thích ý nghĩa của các dòng lệnh

2. Thay lệnh **cats** trong câu 1 bởi lệnh dưới đây, chạy và giải thích kết quả.

```
cats1= pd.cut(ages, [18, 26, 36, 61, 100], right=False)
```

3. Thay lệnh cats trong câu 1 bởi 2 lệnh dưới đây, chạy và giải thích kết quả

```
group_names = ['Youth', 'YoungAdult', 'MiddleAged', 'Senior']
cats2=pd.cut(ages, bins, labels=group_names)
```

Bài 5.

Cho tập dữ liệu new_data được tạo ngẫu nhiên bằng hàm **random.rand()** trong **numpy** như sau.

```
new_data = np.random.rand(20)
```

Thực hiện các công việc như bài 4 với hàm **cut** được cho theo 2 dạng sau :

a/ cats3=pd.cut(new_data,4,precision=2)

b/ cats4=pd.cut(new_data, 4)