

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

KHAI THÁC DỮ LIỆU

**BÀI TẬP LAP 1: BANK**

**NGUYỄN THỊ NGỌC HÀ: 17520421**

Nội dung

[DỮ LIỆU VỀ BANK 3](#_Toc36666400)

[Mô tả các thuộc tính: 3](#_Toc36666401)

[Yêu cầu cơ bản: 3](#_Toc36666402)

[1. Xác định loại thuộc tính: 3](#_Toc36666403)

[2. Xử lý dữ liệu bị thiếu: 3](#_Toc36666404)

[2.1. Đối với dữ liệu số: 3](#_Toc36666405)

[2.2. Đối với dữ liệu khác (Binary, Categorical,..) 5](#_Toc36666406)

[3. Thực hiện khử nhiễu trên thuộc tính Age: 7](#_Toc36666407)

[3.1. Binning: 7](#_Toc36666408)

[3.2. Smoothing: 7](#_Toc36666409)

[4. Rời rạc hóa dữ liệu: 8](#_Toc36666410)

[4.1. Bằng khoảng giá trị: 8](#_Toc36666411)

[4.2. Bằng các khái niệm (Youth, Senior, Adult,..) 9](#_Toc36666412)

[5. Một kiểu biểu diễn khác của dạng dữ liệu ‘contact’ để tránh trường hợp chứa quan hệ cấp bậc: 9](#_Toc36666413)

[6. Thực hiện chuẩn hóa dữu liệu bằng **Min-Max Normalization** 10](#_Toc36666414)

[6.1. Thuộc tính ‘cons.price.idx’: 10](#_Toc36666415)

# ­DỮ LIỆU VỀ BANK

## Mô tả các thuộc tính:

* Job: Nghề nghiệp
* Marital: Tình trạng hôn nhân
* Education: Trình độ học vấn
* Default: Đã có tín dụng mặc định hay chưa
* Loan: Đã có khoản vay nào chưa
* Duration: Thời lượng lần contact cuối cùng, tính trên đơn vị giây
* Cons.price.idx: Chỉ số giá tiêu dùng
* Age: Tuổi

## Yêu cầu cơ bản:

### Xác định loại thuộc tính:

* Job: Nominal Thuộc tính phân loại
* Marital Binary Symmetric: Thuộc tính nhị phân đối xứng
* Education: Ordinal: Thuộc tính xếp hạng
* Default: Binary Asymmetric: Thuộc tính nhị phân bất đối xứng
* Loan: Binary Asymmetric: Thuộc tính nhị phân bất đối xứng
* Duration: Ratio-scaled: Thuộc tính giá trị theo tỉ lệ
* Cons.price.idx: Ratio-scaled: Thuộc tính giá trị theo tỉ lệ
* Age: Ratio-scaled: Thuộc tính giá trị theo tỉ lệ

### Xử lý dữ liệu bị thiếu:

#### Đối với dữ liệu số:

* Xử lý thay thế bằng giá trị trung bình (mean) trên thuộc tính tuổi
* Tính giá trị trung bình:
* Mean = = 40,11 tức xấp xỉ 40
* Điền giá trị này vào dữ liệu Tuổi thiếu
* Xử lý bằng giá trị trung bình của thuộc tính quyết định

Ví dụ: Tính trung bình (mean) của tuổi (age) của các khách hàng có ‘marital’ = ‘married’

* Mean\_married == 42,389 tức xấp xỉ 42
* Điền giá trị 29 cho các khách hàng bị thiếu dữ liệu tuổi có ‘marital’ = ‘married’

Làm tương tự với thuộc tính ‘marital’ = ‘single’ ta có:

* Mean\_single = = 33,209 xấp xỉ 33
* Điền giá trị 33 cho các khách hàng bị thiếu dữ liệu tuổi có ‘marital’ = ‘single’

Làm tương tự với thuộc tính ‘marital’ = ‘divorced’ ta có:

* Mean\_divorced = = 45,1 xấp xỉ 45
* Điền giá trị 45 cho các khách hàng bị thiếu dữ liệu tuổi có ‘marital’ = ‘divorced’

Làm tương tự với thuộc tính ‘marital’ = ‘unknown’ ta có:

* Mean\_unknown = = 42,27 xấp xỉ 42
* Điền giá trị 42 cho các khách hàng bị thiếu dữ liệu tuổi có ‘marital’ = ‘unknown’

#### Đối với dữ liệu khác (Binary, Categorical,..)

* Các cách xử lý
* Loại bỏ mẫu có giá trị thiếu ra khỏi tập dữ liệu
* Điền giá trị thiếu bằng giá trị xuất hiện nhiều nhất (mode) trong thuộc tính.
* Xem thuộc tính thiếu là một loại mới ( đối với categorical)
* Dự đoán giá trị dữ liệu.

##### Thuộc tính ‘marital’ :

Gồm 11 dòng dữ liệu có giá trị ‘unknown’

Do số dòng dữ liệu ít, có thể ít ảnh hưởng đến tập dữ liệu chung, nên ta có thể chọn phương thức loại bỏ 2 dòng dữ liệu này.

Cách khác, ta có thể tìm Mode của thuộc tính này. Ở đây, Mode của thuộc tính ‘marital’ = ‘married’ với 2509 bộ dữ liệu

* Do đó ta có thể thay thế 11 dòng dữ liệu ‘unknown’ thành ‘married’ để điền đầy đủ bộ dữ liệu.

##### Thuộc tính ‘education’:

Gồm 167 dòng dữ liệu có giá trị ‘unknown’

Do số lượng dữ liệu unknown quá nhiều, nếu xóa sẽ gây ảnh hưởng tới bộ dữ liệu.

* Ta có những hướng xử lý sau:
* Tìm giá trị xuất hiện nhiều nhất của thuộc tính quyết định ‘marital’ = ‘married’
* Gồm 95 dòng dữ liệu unknown của thuộc tính ‘education’ đối với thuộc tính ‘marital’ = ‘married’
* Trong đó Mode của thuộc tính ‘education’ = ‘university.degree’ là 677 giá trị.
* Thay thế giá trị ‘university.degree’ vào 95 giá trị ‘unknown’ của thuộc tính ‘education’ đối với thuộc tính quyết định ‘marital’ = ‘married’
* Tìm giá trị xuất hiện nhiều nhất của thuộc tính quyết định ‘marital’ = ‘single’
* Gồm 55 dòng dữ liệu unknown của thuộc tính ‘education’ đối với thuộc tính ‘marital’ = ‘single’
* Trong đó Mode của thuộc tính ‘education’ = ‘university.degree’ là 450 giá trị.
* Thay thế giá trị ‘university.degree’ vào 55 giá trị ‘unknown’ của thuộc tính ‘education’ đối với thuộc tính quyết định ‘marital’ = ‘single’
* Tìm giá trị xuất hiện nhiều nhất của thuộc tính quyết định ‘marital’ = ‘divorced’
* Gồm 17 dòng dữ liệu unknown của thuộc tính ‘education’ đối với thuộc tính ‘marital’ = ‘divorced’
* Trong đó Mode của thuộc tính ‘education’ = ‘university.degree’ là 132 giá trị.
* Thay thế giá trị ‘university.degree’ vào 17 giá trị ‘unknown’ của thuộc tính ‘education’ đối với thuộc tính quyết định ‘marital’ = ‘divorced’
* Xem thuộc tính thiếu là một loại mới

Do số giá trị = ‘unknown’ có 167 giá trị, ta có thể xem đây là một giá trị dữ liệu thứ 4 của thuộc tính mà không cần loại bỏ đi, hay tìm cách thay thế chúng.

### Thực hiện khử nhiễu trên thuộc tính Age:

#### Binning:

Để Binning (chia giỏ) ta lấy giá trị min và giá trị max của thuộc tính là giá trị bắt đầu và kết thúc cho mỗi giỏ. Chọn độ rộng mỗi giỏ là 25, ta được các giỏ sau: (0,25], (25,50], (50,75], (75,100). Tiếp theo, đem dữ liệu phân vào các giỏ đã chia.

Như vậy, giá trị min của thuộc tính tuổi (Age) theo bảng số liệu đã lọc là 1, giá trị max là 80.

Ta chia thành các giỏ tương tự như hướng dẫn, các giá trị trong các giỏ tương ứng là:

+ Giỏ (0,25] gồm 155 giá trị

+ Giỏ (25,50] gồm 3217 giá trị

+ Giỏ (50,75] gồm 725 giá trị

+ Giỏ (75,100] gồm 22 giá trị

#### Smoothing:

Để Smoothing ta tính giá trị trung bình của các giá trị trong cùng một giỏ, sau đó thay thế giá trị trung bình vừa tính được vào các giá trị đang hiện hữu trong giỏ.

+ Giỏ (0,25] gồm 155 giá trị

* Mean(0,25) = = 23,75 xấp xỉ 24
* Điền giá trị trung bình = 24 vừa tìm được thay cho các giá trị đang hiện hữu trong giỏ.

+ Giỏ (25,50] gồm 3217 giá trị

* Mean(25,50] = = 37,077 xấp xỉ 37
* Điền giá trị trung bình = 37 vừa tìm được thay cho các giá trị đang hiện hữu trong giỏ.

+ Giỏ (50,75] gồm 725 giá trị

* Mean(50,75] = = 55,87 xấp xỉ 56
* Điền giá trị trung bình = 56 vừa tìm được thay cho các giá trị đang hiện hữu trong giỏ.

+ Giỏ (75,100] gồm 22 giá trị

* Mean(75,100] = = 80,18 xấp xỉ 80
* Điền giá trị trung bình = 80 vừa tìm được thay cho các giá trị đang hiện hữu trong giỏ.

### Rời rạc hóa dữ liệu:

#### Bằng khoảng giá trị:

Các khoảng giá trị là:

* Giỏ 1 [0,10): 1,2,3,4,5,6,7,8,9
* Giỏ 2 [10,20): 10,11,12,13,14,15,16,17,18,19
* Giỏ 3 [20,30): 20, 21, 22, 23, 24, 25, 26, 27, 28, 29
* Giỏ 4 [30,40): 30,31,32,33,34,35,36,37,38,39
* Giỏ 5 [40,50): 40, 41,42,43,44,45,46,47,48,49
* Giỏ 6 [50,60): 50,51,52,53,54,55,56,57,58,59
* Giỏ 7 [60,70): 60,61,62,63,64,65,66,67,68,69
* Giỏ 8 [70,80): 70,71,72,73,74,75,76,77,78
* Giỏ 9 [80,90): 80,81,82,85,86,88

#### Bằng các khái niệm (Youth, Senior, Adult,..)

* **Kid**: (0,10]: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
* **Youth** (10,18]: 11, 12, 13, 14, 15, 16, 17, 18
* **Young Adult** (18,29]: 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29
* **Adult** (29,55]: 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55
* **Senior** (>55): 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 80, 81, 82, 85, 86, 88

### Một kiểu biểu diễn khác của dạng dữ liệu ‘contact’ để tránh trường hợp chứa quan hệ cấp bậc:

Để đảm bảo công bằng cho các thuộc tính định tính, ta có thể biểu diễn thuộc tính này thành dạng One-hot Encoding. Ở dạng biểu diễn này, mỗi giá trị của thuộc tính được biểu diễn bằng một vector với một thành phần có giá trị là 1 và các thành phần còn lại có giá trị là 0. Số lượng thành phần của vector chính là số loại của thuộc tính.

Ở đây, thuộc tính ‘contact’ có 2 loại: cellular, telephone

Như vậy tương ứng với từng loại, ta sẽ biểu diễn One-hot như sau:

* cellular: 1,0
* telephone: 0,1

### Thực hiện chuẩn hóa dữu liệu bằng **Min-Max Normalization**

Khi sử dụng dữ liệu cho các thuật toán phân lớp hoặc gom cụm (k-NN, Neural Networks, k-Means,..) để tránh tình trạng các thuộc tính nằm trong vùng giá trị lớn hơn có xu hướng ảnh hưởng đến mô hình nhiều hơn các dữ liệu nằm trong vùng có giá trị nhỏ hơn (Ví dụ: Tuổi 20, thu nhập 4.000.000). Ta thực hiện việc chuẩn hóa các thuộc tính về một vùng giá trị.

Đối với Min-Max Normalization, ta chuyển dữ liệu về khoảng giá trị thuộc vùng 0 – 1 bằng công thức sau:

new\_value =

Trong đó:

* new\_value = Gía trị sau khi chuyển đổi
* value = Gía trị đang cần chuyển đổi
* min = Gía trị thấp nhất của thuộc tính
* max = Gía trị lớn nhất của thuộc tính

#### Thuộc tính ‘cons.price.idx’:

Xét gái trị 94,199 của thuộc tính Chỉ số tiêu dùng ‘cons.price.idx’. Áp dụng công thức trên ta tính được giá trị mới là: 0,7786

Tương tự với các giá trị còn lại, ta sẽ chuẩn hóa được các giá trị về một vùng dữ liệu nhất định.