

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

KHAI THÁC DỮ LIỆU

**BÀI TẬP LAP 1: TITANIC**

**NGUYỄN THỊ NGỌC HÀ: 17520421**

Nội dung

[DỮ LIỆU VỀ TITANIC 3](#_Toc36654247)

[Mô tả các thuộc tính: 3](#_Toc36654248)

[Yêu cầu cơ bản: 3](#_Toc36654249)

[1. Xác định loại thuộc tính: 3](#_Toc36654250)

[2. Xử lý dữ liệu bị thiếu: 3](#_Toc36654251)

[2.1. Đối với dữ liệu số: 3](#_Toc36654252)

[2.2. Đối với dữ liệu khác (Binary, Categorical,..) 6](#_Toc36654253)

[3. Thực hiện khử nhiễu trên thuộc tính Age: 8](#_Toc36654254)

[3.1. Binning: 8](#_Toc36654255)

[3.2. Smoothing: 8](#_Toc36654256)

[4. Rời rạc hóa dữ liệu: 9](#_Toc36654257)

[4.1. Bằng khoảng giá trị: 9](#_Toc36654258)

[4.2. Bằng các khái niệm (Youth, Senior, Adult,..) 10](#_Toc36654259)

[5. Một kiểu biểu diễn khác của dạng dữ liệu ‘Embarked’ để tránh trường hợp chứ quan hệ cấp bậc: 10](#_Toc36654260)

[6. Thực hiện chuẩn hóa dữu liệu bằng **Min-Max Normalization** 11](#_Toc36654261)

[6.1. Thuộc tính ‘Fare’: 11](#_Toc36654262)

# ­DỮ LIỆU VỀ TITANIC

## Mô tả các thuộc tính:

* Survival: Hành khách sống sót hay không (Yes: 1, No: 0). Survival là thuộc tính quyết định.
* Pclass: Loại vé ( 1: 1st, 2: 2nd, 3: 3rd).
* Name: Tên hành khách.
* Sex: Giới tính.
* Age: Tuổi.
* Sibsp: Số vợ hoặc chồng / anh chị em trên tàu.
* Parch: Số lượng cha mẹ / con cái trên tàu.
* Embarked: Cổng lên tàu.

## Yêu cầu cơ bản:

### Xác định loại thuộc tính:

* Survival: Binary Asymmetric: Thuộc tính nhị phân bất đối xứng
* Pclass: Ordinal: Thuộc tính xếp hạng
* Name: Nominal: Thuộc tính phân loại
* Sex: Binary Symmetric: Thuộc tính nhị phân đối xứng
* Age: Ratio-scaled: Thuộc tính giá trị theo tỉ lệ
* Sibsp: Ratio-scaled: Thuộc tính giá trị theo tỉ lệ
* Parch: Ratio-scaled: Thuộc tính giá trị theo tỉ lệ
* Embarked Nominal: Thuộc tính phân loại

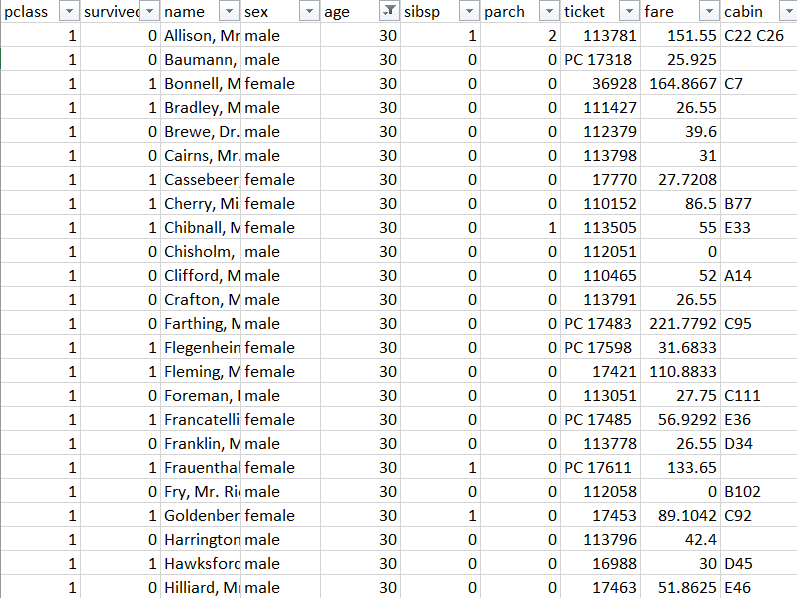
### Xử lý dữ liệu bị thiếu:

#### 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 = =29,881 tức xấp xỉ 30

* Điền giá trị này vào dữ liệu Tuổi thiếu ta được bảng sau:

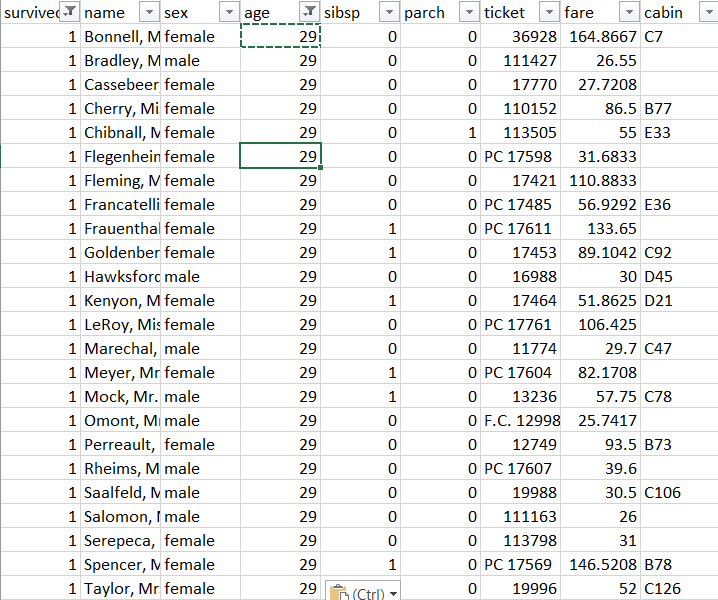


* Xử lý bằng giá trị trung bình của thuộc tính quyết định

Ví dụ: Tính trung bình (mean) của tuổi (age) của các hành khách có ‘survived’ = 1

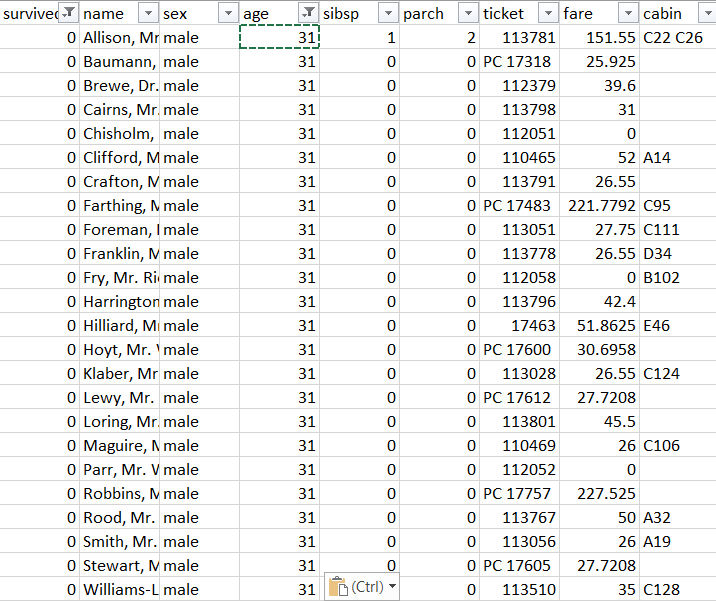
Mean1 == 28,87 tức xấp xỉ 29

* Điền giá trị 29 cho các hành khách bị thiếu dữ liệu tuổi có ‘survived’=1 ta được bảng sau:



Làm tương tự với thuộc tính ‘survived’ = 0

* Mean0 = = 30,568 xấp xỉ 31
* Điền giá trị 31 cho các hành khách bị thiếu dữ liệu tuổi có ‘survived’ = 0 ta được bảng sau:



#### 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 ‘Embarked’ :

Gồm 2 dòng dữ liệu trống

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 ‘Embarked’ = ‘S’

* Do đó ta có thể thay thế 2 dòng dữ liệu bị thiếu thành ‘S’ để điền đầy đủ bộ dữ liệu.

##### Thuộc tính ‘Home.dest’:

Gồm 564 dòng dữ liệu trống

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

* Ta có 2 hướng xử lý sau:
* Tìm giá trị xuất hiện nhiều nhất của thuộc tính quyết định ‘survived’ = 1
* Gồm 156 dòng dữ liệu trống đối với thuộc tính ‘survived’ = 1
* Trong đó Mode của thuộc tính ‘home.dest’ = ‘New York, NY’ và các giá trị liên quan ‘New York, NY’ là 76 giá trị.
* Thay thế giá trị ‘New York, NY’ vào các giá trị bị thiếu của thuộc tính ‘home.dest’ đối với thuộc tính quyết định ‘survived’ = 1
* Tìm giá trị xuất hiện nhiều nhất của thuộc tính quyết địng ‘survived’ = 0
* Gồm 398 dòng dữ liệu trống đối với thuộc tính ‘survived’ = 0

Trong đó Mode của thuộc tính ‘home.dest’ = ‘England’ và các giá trị liên quan ‘England’ là 59 giá trị.

* Thay thế giá trị ‘England’ vào các giá trị bị thiếu của thuộc tính ‘home.dest’ đối với thuộc tính quyết định ‘survived’ = 0

### 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 20, ta được các giỏ sau: (0,20), (20,40), (40,60), (60,80). 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 (0,20) gồm 213 giá trị

+ Giỏ (20,40) gồm 839 giá trị

+ Giỏ (40,60) gồm 205 giá trị

+ Giỏ (60,80) gồm 40 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 (0,20) gồm 213 giá trị

* Mean(0,20) = = 12,685 xấp xỉ 13
* Điền giá trị trung bình = 13 vừa tìm được thay cho các giá trị đang hiện hữu trong giỏ.

+ Giỏ (20,40) gồm 839 giá trị

* Mean(20,40) = = 28,78 xấp xỉ 29
* Điền giá trị trung bình = 29 vừa tìm được thay cho các giá trị đang hiện hữu trong giỏ.

+ Giỏ (40,60) gồm 205 giá trị

* Mean(40,60) = = 47,366 xấp xỉ 47
* Điền giá trị trung bình = 47 vừa tìm được thay cho các giá trị đang hiện hữu trong giỏ.

+ Giỏ (60,80) gồm 40 giá trị

* Mean(60,80) = = 64,45 xấp xỉ 64
* Điền giá trị trung bình = 64 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): 10,11,11.5,12,13,14,14.5,15,16,17,18,18.5,19
* Giỏ 3 [20,30): 20, 20.5, 21, 22, 22.5, 23, 23.5, 24, 24.5, 25, 26, 26.5, 27, 28, 28.5, 29
* Giỏ 4 [30,40): 30,30.5,31,32,32.5,33,34,34.5,35,36,36.5,37,38,38.5,39
* Giỏ 5 [40,50): 40,40.5,41,42,43,44,45,45.5,46,47,48,49
* Giỏ 6 [50,60): 50,51,52,53,54,55,55.5,56,57,58,59
* Giỏ 7 [60,70): 60,60.5,61,62,63,64,65,66,67
* Giỏ 8 [70,80]: 70,70.5,71,74,76,80

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

* **Kid**: (0,10]: 1,2,3,4,5,6,7,8,9,10
* **Youth** (10,18]: 11,11.5,12,13,14,14.5,15,16,17,18,18.5,19, 20, 20.5, 21, 22, 22.5, 23, 23.5, 24, 24.5, 25, 26, 26.5, 27, 28, 28.5, 29
* **Young Adult** (18,29]: ,18.5,19, 20, 20.5, 21, 22, 22.5, 23, 23.5, 24, 24.5, 25, 26, 26.5, 27, 28, 28.5, 29
* **Adult** (29,55]: 30,30.5,31,32,32.5,33,34,34.5,35,36,36.5,37, 38, 38.5, 39, 40, 40.5, 41, 42, 43, 44, 45, 45.5, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55
* **Senior** (>55): 55.5, 56, 57, 58, 59, 60, 60.5, 61, 62, 63, 64, 65, 66, 67, 70, 70.5, 71, 74, 76, 80

### Một kiểu biểu diễn khác của dạng dữ liệu ‘Embarked’ để tránh trường hợp chứ 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 ‘Embarked’ có 3 loại: Q, C, S

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

* Q: 1,0,0
* C: 0,1,0
* S: 0,0,1

### 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 ‘Fare’:

Xét gái trị 7.25 của thuộc tính giá vé ‘Fare’. Áp dụng công thức trên ta tính được giá trị mới là: 0,00039

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