ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỌC MÁY TÍNH

KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG – CS313.N22

Bài tập 2: Cách sử dụng ngôn ngữ Python trong quá trình chuẩn bị dữ liệu

Nhóm 3:

Bùi Nguyễn Anh Trung (NT)	20520332
Hồ Thanh Tịnh	20520813
Nguyễn Trần Minh Anh	20520394
Lê Nguyễn Bảo Hân	20520174
Nguyễn Văn Đức Ngọc	20521666

Hồ Chí Minh, 13 tháng 03 năm 2023

Nội dung

Bảng phân công

Thành viên / Công Việc	Bùi Nguyễn Anh Trung	Hồ Thanh Tịnh	Nguyễn Trần Minh Anh	Lê Nguyễn Bảo Hân	Nguyễn Văn Đức Ngọc
Phân công, quản lý công việc chung	✓				
Tim hiểu nội dung làm sạch dữ liệu		✓			
Tim hiểu nội dung tích hợp dữ liệu			✓		
Tổng hợp nội dung báo cáo Định dạng báo cáo			✓	✓	
Chuẩn bị demo Thuyết trình bài tập 1	✓				
Tìm hiểu nội dung chuẩn hóa dữ liệu				✓	
Tìm hiểu nội dung làm giảm dữ liệu					✓
Làm slide		✓		✓	

Đóng góp, chỉnh sửa và hoàn thiện nội dung	✓				
Mức độ hoàn thiện (%)	100%	100%	100%	100%	100%

I. Tổng quan về chuẩn bị dữ liệu

1. Tại sao cần chuẩn bị dữ liệu?

Trên thực tế, dữ liệu thường có kích thước rất lớn và được tổng hợp từ nhiều nguồn không đồng nhất, dẫn đến cơ sở dữ liệu thường gặp các tình trạng [1] [2]:

- Không đầy đủ: thiếu giá trị thuộc tính hoặc một vài thuộc tính quan tâm nhất định, hoặc chỉ chứa dữ liệu tổng hợp
- Không chính xác hoặc nhiễu: chứa lỗi và giá trị sai lệch so với kỳ vọng
- Không nhất quán: dữ liệu tồn tại ở các định dạng khác nhau trong nhiều bảng

Chất lượng dữ liệu thấp sẽ dẫn đến kết quả khai thác kém hiệu quả. Vậy, *"Làm thế nào để tiền xử lý dữ liệu, từ đó, cải thiện chất lượng dữ liệu và kết quả khai thác? Làm thế nào để tiền xử lý dữ liệu để khiến cho quá trình khai thác dễ dàng hơn?"*

Việc áp dụng các kỹ thuật chuẩn bị dữ liệu có thể cải thiện đáng kể chất lượng các mẫu và thời gian cần thiết cho quá trình khai thác thực tế.

Dữ liệu được đánh giá là chất lượng nếu đáp ứng được các yêu cầu của mục đích sử dụng. Có nhiều yếu tố cấu thành chất lượng dữ liệu, bao gồm *tính*

chính xác, *tính đầy đủ*, *tính nhất quán*, *kịp thời*, *đáng tin cậy* và *có thể diễn giải*.

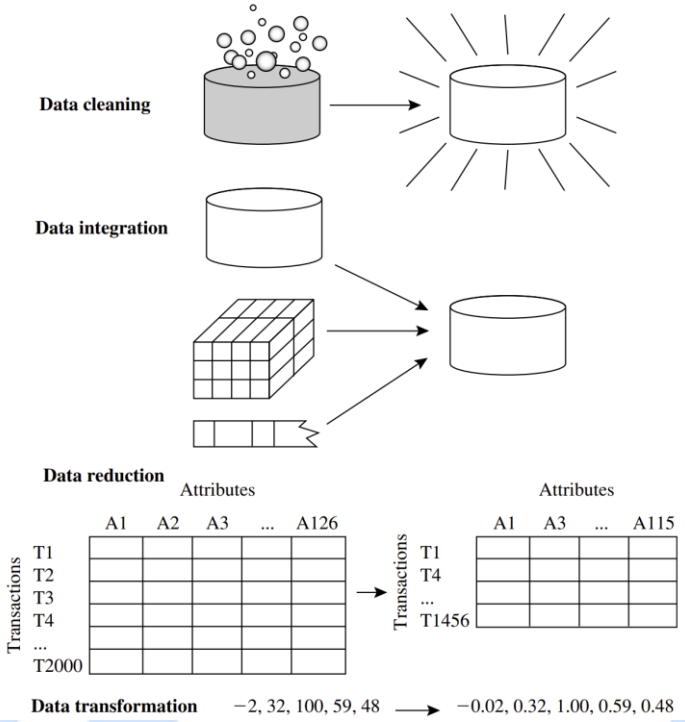
Có nhiều lý do dẫn đến chất lượng dữ liệu không đạt chuẩn:

- **Tính chính xác:** đúng hay sai, chính xác hay không
    - Lỗi khi nhập liệu
    - Cố tình nhập sai giá trị cho các trường bắt buộc khi không muốn gửi thông tin cá nhân
    - Quy ước đặt tên hoặc mã không nhất quán
- **Tính đầy đủ:** không được điền, rỗng,…
    - Các thuộc tính quan tâm có thể không có sẵn
    - Dữ liệu không được thêm vào vì không được xem là quan trọng trong thời gian đầu
    - Dữ liệu liên quan không được ghi nhận do hiểu lầm
    - Thiếu dữ liệu, các bộ thiếu giá trị thuộc tính có thể cần được bao hàm
- **Tính nhất quán:** chỉ một số được thay đổi, thiếu chặt chẽ,…
    - Mã dữ liệu hoặc định dạng không nhất quán
- **Kịp thời:** mức độ cập nhật
    - Dữ liệu có được cập nhật kịp thời không?
- **Đáng tin cậy:** mức độ tin cậy độ chính xác dữ liệu
    - Mức độ tin cậy của dữ liệu đối với người dùng?
    - Các lỗi trước có thể ảnh hưởng đến độ tin cậy dữ liệu
- **Có thể diễn giải:** mức độ dễ hiểu của dữ liệu

2. Các tác vụ chính trong chuẩn bị dữ liệu

Các tác vụ chính trong chuẩn bị dữ liệu bao gồm:

- **Làm sạch dữ liệu:** loại bỏ nhiễu và sửa lỗi không nhất quán trong dữ liệu
    - Điền vào các giá trị còn thiếu
    - Làm trơn dữ liệu, xác định hoặc loại bỏ các giá trị ngoại lệ
    - Giải quyết các điểm không nhất quán
- **Tích hợp dữ liệu:** hợp nhất dữ liệu từ nhiều nguồn vào một kho lưu trữ dữ liệu nhất quán
    - Tích hợp nhiều cơ sở dữ liệu, khối dữ liệu hoặc tệp
    - Loại bỏ dữ liệu dư thừa và trùng lặp
    - Phát hiện và giải quyết mâu thuẫn trong dữ liệu
- **Rút gọn dữ liệu:** tổng hợp, loại bỏ các đặc trưng dư thừa hoặc phân cụm
    - Tổng hợp và tổng quát hóa
    - Giảm chiều dữ liệu
    - Giảm số lượng
    - Nén dữ liệu
- **Biến đổi dữ liệu:** biến đổi dữ liệu thành dạng phù hợp và thuận tiện cho thuật toán khai thác dữ liệu
    - Chuẩn hóa
    - Rời rạc hóa
    - Phân cấp khái niệm



Hình 1. Quá trình chuẩn bị dữ liệu

II. Quá trình chuẩn bị dữ liệu

1. Làm sạch dữ liệu

1.1 Khái niệm

Là một quá trình làm "sạch" dữ liệu bằng các phương pháp như:

- Điền dữ liệu thiếu
- Làm "mịn" dữ liệu nhiễu
- Xác định và xóa đi các dữ liệu ngoại biên
- Giải quyết các vấn đề không nhất quán dữ liệu

Hình 2. Minh họa làm sạch dữ liệu

1.2 Tại sao phải làm sạch dữ liệu?

Trên thực tế, dữ liệu thường không "sạch". Có rất nhiều lý do khiến dữ liệu bị sai:

- Lỗi do thiết bị nhập
- Lỗi con người
- Lỗi tính toán của máy tính
- Lỗi trong quá trình truyền dữ liệu

Từ đó dẫn đến dữ liệu sẽ gặp vấn đề sau:

- Chưa hoàn thiện : Thiếu giá trị thuộc tính, thiếu thuộc tính
    - *Ví dụ: phone_number = "" (bỏ trống)*
- Dữ liệu bị nhiễu : Chứa nhiều giá trị gây nhiễu, giá trị lỗi, giá trị ngoại biên
    - *Ví dụ: weight = "-50" (giá trị lỗi, do cân nặng thì không âm)*
- Không nhất quán : Không thỏa tính logic
    - *Ví dụ: Age = "60" , Birthday="02/05/2002" (Tính đến 2023 thì tuổi lớn nhất chỉ là 21 tuổi); Đang đánh giá bằng "1,2,3" nay đổi sang "A,B,C"*
- Dữ liệu sai do cố tình: Thường là do xử lý không tốt các trường dữ liệu thiếu
    - *Ví dụ: Tạo thêm trường dữ liệu ngày sinh và khởi tạo tất cả cùng sinh ngày 1/1/2000*

Nếu người dung không tin dữ liệu "sạch sẽ", họ sẽ dường như không tin vào kết quả của quá trình phân tích dữ liệu. Vì dữ liệu không tốt sẽ cho ra những kết quả phân tích không đáng tin cậy (Garbage In Garbage Out). Vậy nên cần làm sạch dữ liệu.

1.3 Các phương pháp làm sạch dữ liệu

1.3.1 Điền các giá trị còn thiếu

Vấn đề

Giá trị thu thập bị thiếu hoặc sai định dạng ở một số thuộc tính (Unknown: là không xác định, NULL: bỏ trống)

Bảng 1. Bảng dữ liệu chứa giá trị không xác định hoặc bỏ trống

NAME	PLACE	PHONE
Tom	USA	250684737
John	UAE	**NULL**
Andy	**Unknown**	286839206

Phương pháp

Có 2 phương pháp chính:

- Bỏ qua các mẫu tin có giá trị thiếu:

  Thường dùng khi thiếu nhãn của lớp (trong phân lớp). Phương pháp nhanh chóng và dễ dàng nhất, nhưng không hiệu quả vì có thể sẽ loại bỏ hoặc mất thông tin đặc biệt khi tỷ lệ giá trị thiếu của thuộc tính.

Bảng 2. Bảng dữ liệu chứa mẫu tin thiếu giá trị

Patient ID	Body temperature	WBC	Sputum culture	Chest X-ray
A001	Normal	Normal	Negative	Normal
A002	Fever	High	Negative	Infiltrate
A003	Normal	Normal	Negative	Normal
A004	Fever	High	Positive	Infiltrate
A005	Normal	Normal	Negative	(Missing)
⋮				

- Điền các giá trị thiếu tự động bằng:
  - Giá trị trung bình của tất cả thuộc tính.
  - Giá trị trung bình hoặc trung vị của thuộc tính trong từng lớp.
  - Giá trị có nhiều khả năng nhất: suy ra từ công thức Bayesian, cây quyết định, KNN
  - Nhập các giá trị thiếu dựa trên các quan sát khác

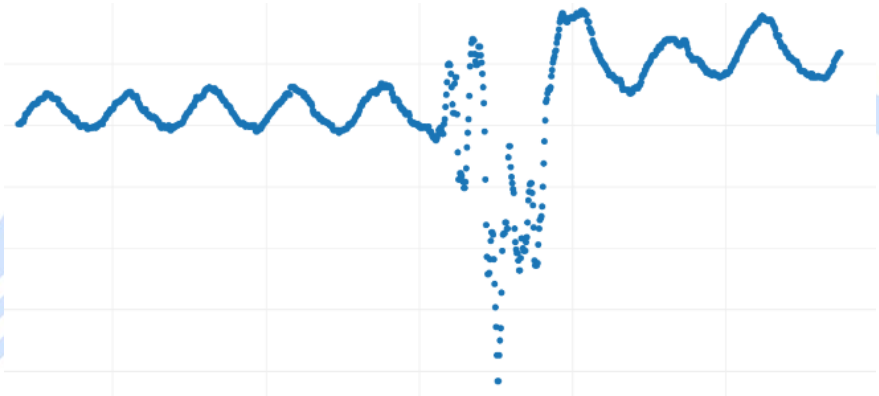
  Phương pháp này có thể dẫn đến việc làm mất tính toàn vẹn của dữ liệu vì dữ liệu được thêm vào không phải dữ liệu thực tế.

1.3.2 Xác định các sai biệt và khử dữ liệu tạp, nhiễu

Vấn đề

Dữ liệu nhiễu bao gồm dữ liệu vô nghĩa, dữ liệu bị hỏng, dữ liệu nào mà máy móc không thể hiểu và diễn giải một cách chính xác.

Biến số bị lỗi do hàm tính toán, do các hàm ngẫu nhiên hay do chủ đích của người tạo bộ dữ liệu.



Hình 3. Minh họa dữ liệu nhiễu

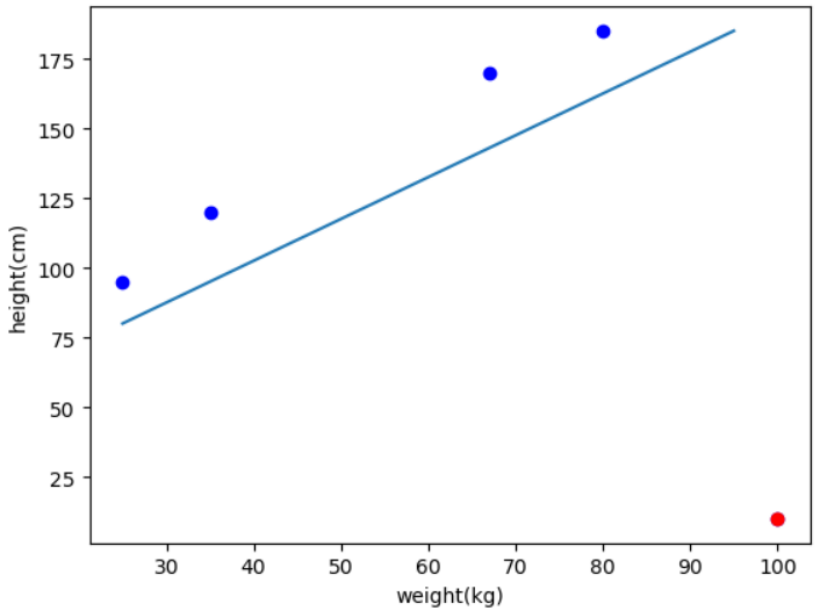
Phương pháp

1. **Hồi quy**

   Phương pháp này sử dụng mô hình hồi qui để dự đoán các giá trị của biến phụ thuộc dựa trên các biến độc lập.

   Các bước thực hiện:

   - Xác định biến phụ thuộc và các biến độc lập để tạo mô hình hồi qui.
   - Kiểm tra độ tin cậy và độ chính xác của mô hình
   - Sử dụng mô hình vừa tạo để dự đoán
   - Xác định các giá trị nhiễu: So sánh các giá trị dự đoán với các giá trị thực tế của biến phụ thuộc
   - Loại bỏ các giá trị nhiễu



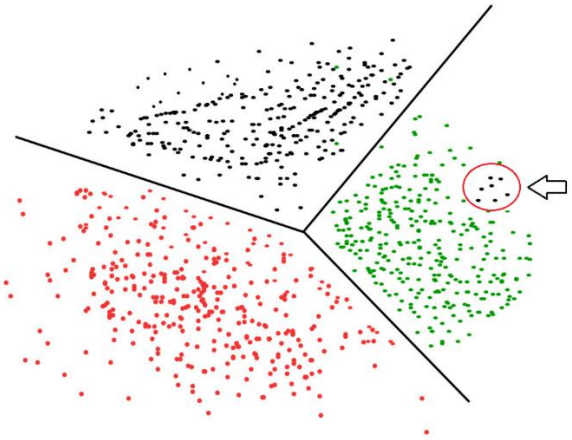
Hình 4. Biểu đồ dữ liệu cân nặng và chiều cao ở người

2. Gom nhóm

Phương pháp gom nhóm cũng tương tự như phương pháp hồi qui, có thể sử dụng các mô hình, kĩ thuật gom nhóm để loại bỏ nhiễu.

Các bước thực hiện:

- Xác định độ tương đồng, quy luật, đặc trưng, điều kiện để gom nhóm
- Kiểm tra và đánh giá từng phương pháp gom nhóm
- Sử dụng phương pháp gom nhóm tối ưu nhất
- Xác định các giá trị nhiễu: Các dữ liệu có giá trị có thể thuộc nhiều nhóm một lúc hoặc không thuộc nhóm nào
- Loại bỏ các giá trị nhiễu



Hình 5. Minh họa gom nhóm dữ liệu

3. Chia giỏ

Hình 6. Ý tưởng phương pháp chia giỏ

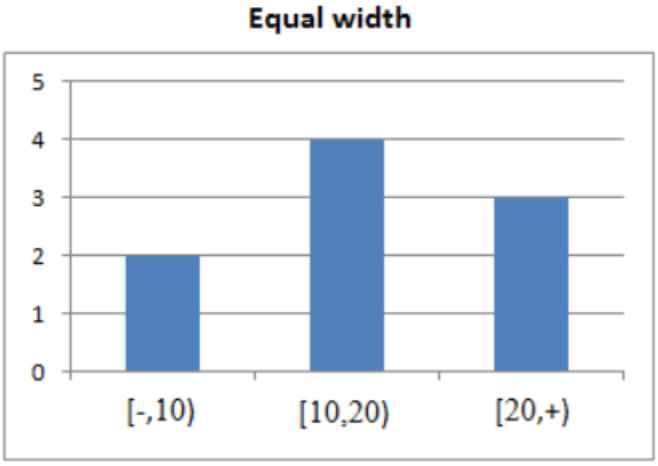
- **Chia theo độ rộng:** *Chia các giỏ sao cho độ rộng bằng nhau mà không quan tâm tới giá trị trong giỏ*

  Chia vùng giá trị thành N khoảng cùng kích thước bin:

$$bin = \frac{(max - min)}{N}$$

  Trong đó:

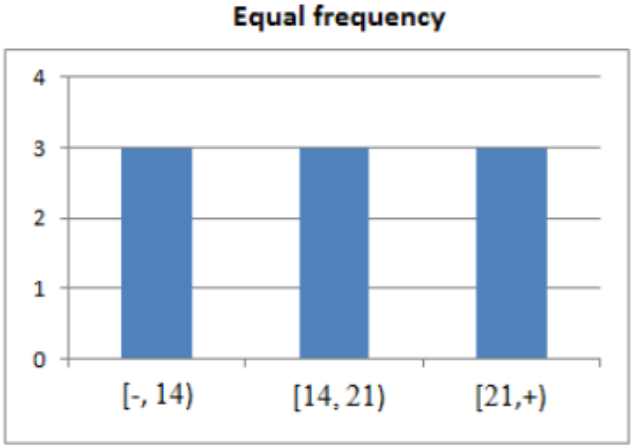
  - max, min: giá trị lớn nhất, giá trị nhỏ nhất
  - N: số khoảng
  - bin: độ rộng mỗi khoảng



Hình 7. Chia theo độ rộng

- **Chia theo độ sâu:** *Chia sao cho giá trị trong giỏ là bằng nhau mà không quan tâm tới độ rộng giỏ*

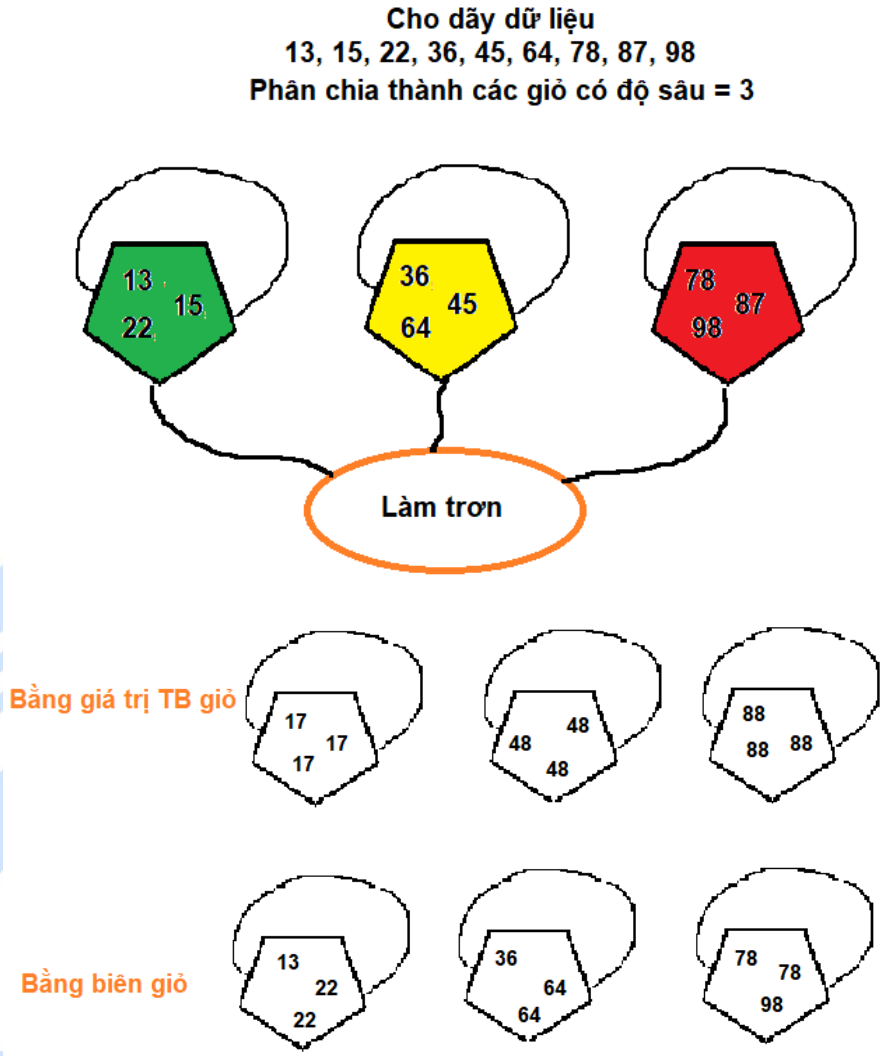
  Chia vùng giá trị thành N khoảng mà mỗi khoảng có chứa gần như cùng số lượng mẫu.



Hình 8. Chia theo độ sâu

- **Làm trơn:** *Sau mỗi phương pháp chia giỏ, dl thường sẽ được làm trơn*

  Có thể làm trơn bằng giá trị trung bình, trung tuyến, biên giỏ,...



Hình 9. Ví dụ chia giỏ và làm trơn dữ liệu

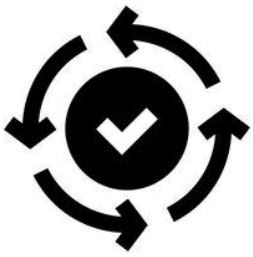
1.3.3 Chỉnh sữa các dữ liệu mâu thuẫn

Vấn đề

Mâu thuẫn dữ liệu (Data Inconsistency) là tình trạng dữ liệu không nhất quán trong cùng một tập dữ liệu, có thể do dữ liệu bị lỗi, bị thiếu hoặc không phù hợp với các quy tắc, ràng buộc hoặc giá trị kỳ vọng. Việc mâu thuẫn dữ liệu có thể ảnh hưởng đến quá trình phân tích dữ liệu và đào tạo mô hình học máy.

Mâu thuẫn dữ liệu thường dễ nhận biết ở các dạng dữ liệu cụ thể, ngắn, có cấu trúc hoặc dữ liệu logic, liên tục. Các dạng mâu thuẫn dữ liệu thường gặp:

- *Mâu thuẫn về định dạng dữ liệu:* Dữ liệu trong các tập tin hoặc cơ sở dữ liệu có thể không có cùng định dạng, ví dụ như ngày tháng được lưu trữ dưới dạng chuỗi (string) trong một bảng và dưới dạng ngày tháng trong bảng khác.
- *Mâu thuẫn về đơn vị đo lường:* Dữ liệu có thể được lưu trữ ở các đơn vị khác nhau, ví dụ như meter và feet.
- *Mâu thuẫn về chính tả:* Dữ liệu có thể được nhập sai hoặc không theo đúng quy tắc chính tả
- *Mâu thuẫn về định danh:* Dữ liệu có thể bị trùng lặp hoặc không có định danh đầy đủ, dẫn đến khó khăn trong việc phân tích và quản lý dữ liệu.
- *Mâu thuẫn về phạm vi hoặc giới hạn:* Dữ liệu có thể không phù hợp với mục đích sử dụng, ví dụ như dữ liệu quá cũ hoặc không đáp ứng được yêu cầu về phạm vi hoặc giới hạn.

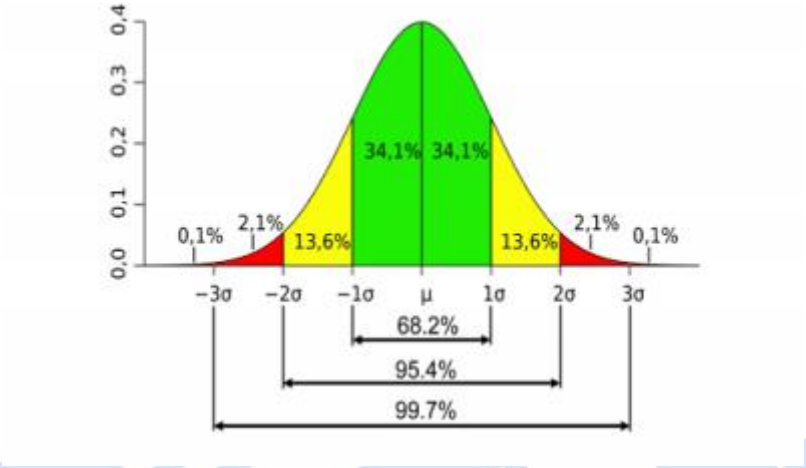


Hình 10. Minh họa Data Consistent

Phương pháp

- Sử dụng các phương pháp biến đổi dữ liệu: chuyển đổi định dạng (ví dụ: từ string sang int), chuyển đổi đơn vị đo lường
- Sử dụng các bộ lọc hoặc các biến điều kiện để loại bỏ hoặc điều chỉnh dữ liệu để chúng rơi vào phạm vi chấp nhận được, hoặc sử dụng các phân phối xác suất

Ví dụ: phân phối Gaussian, phân phối mũ,…



Hình 11. Phân phối Gaussian

Các bước áp dụng Gaussain kernel

Điều kiện:

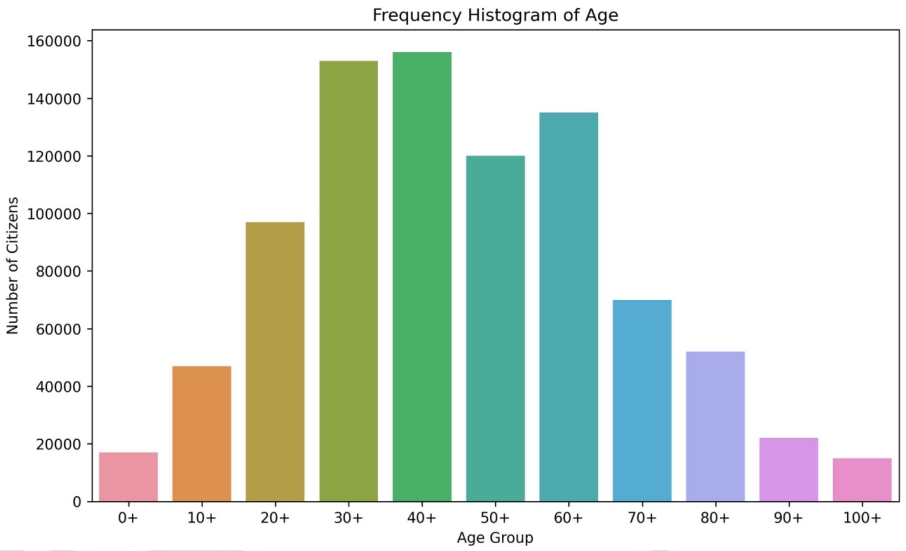
- Thường áp dụng cho dữ liệu tuyến tính 1 chiều và liên tục. Đối với dữ liệu đa chiều, ta thường tính phân phối cho từng đặc trưng (cột) một cách riêng biệt.
- Tập dữ liệu phải có phân phối Gaussian, các tập dữ liệu có kích thước lớn thường có phân phối Gaussian hoặc kể cả khi phân phối đó không đủ điều kiện để là phân phối chuẩn, nhưng vẫn đủ tính chất để ta có thể áp dụng phương pháp này qua các bước bên dưới.

Các bước áp dụng phân phối Gaussian để lọc dữ liệu

- Kiểm tra các điểm dữ liệu có độ tin cậy thấp (các điểm dữ liệu nằm trong vùng đỏ của *Hình 11*)
- Đặt ngưỡng, điều kiện: Dựa trên kết quả kiểm tra các điểm thuộc vùng đỏ, ta có thể đặt ngưỡng để loại bỏ hoàn toàn toàn bộ dữ liệu có độ tin cậy thấp.

Tuy nhiên, đối với hầu hết các tập dữ liệu, các dữ liệu có độ tin cậy thấp này thường rất quan trọng, vì nó thể hiện sự đa dạng, tính toàn

vẹn của tập dữ liệu, đây cũng là nhược điểm của phương pháp Gaussian kernal.

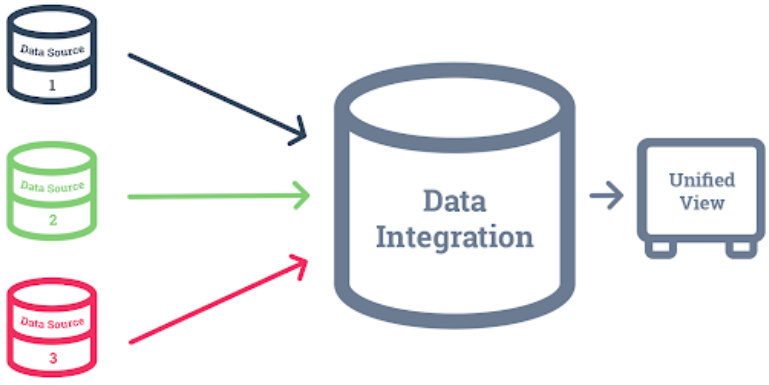


Hình 12. Phân phối mật độ dân số dựa trên số tuổi

4. Tích hợp dữ liệu

2.1 Khái niệm

Tích hợp dữ liệu là quá trình kết hợp dữ liệu không đồng nhất từ các nguồn khác nhau thành một sơ đồ tập hợp dữ liệu duy nhất để từ đó người dùng có thể truy vấn và có một cái nhìn thống nhất về chúng.



Hình 13. Minh họa tích hợp dữ liệu

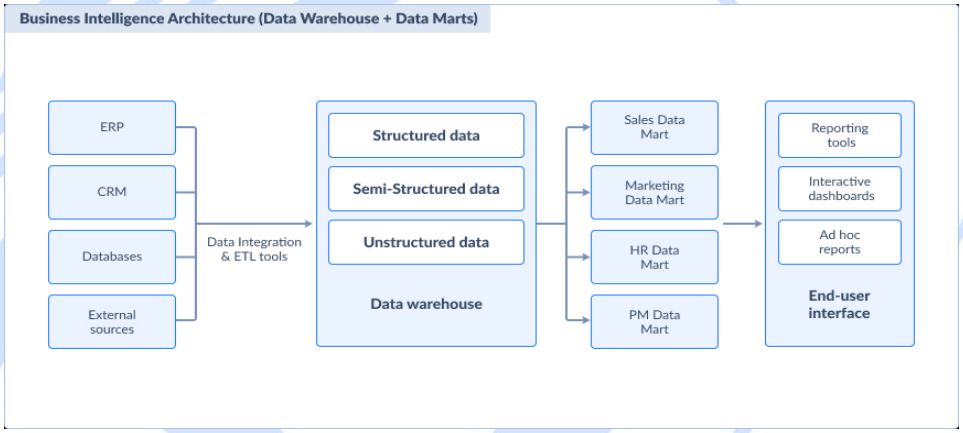
Ví dụ: Marketing là lĩnh vực quảng cáo dựa trên đối tượng cụ thể. Nếu như không được quản lý chặt chẽ, các thông điệp sẽ không thể truyền tới đúng đối tượng vào thời điểm cần thiết. Bất trắc trong việc quản lý thông tin của

hàng triệu khách hàng có thể dẫn tới việc lãng phí công sức, nguồn vốn, thời gian. Tích hợp dữ liệu là cách tốt nhất để giữ cho dữ liệu dễ quản lý, chính xác và hợp thời.

2.2 Tại sao phải tích hợp dữ liệu?

Lợi ích của việc tích hợp dữ liệu trong thực tế

- Tăng độ chính xác cho cơ sở dữ liệu.
- Đưa ra kết luận, phán đoán, phương hướng đáng tin cậy dựa trên thông tin dữ liệu cung cấp.
- Hiệu suất cao, việc tích hợp dữ liệu sẽ giúp tinh giản, loại bỏ dữ liệu dư thừa, đồng thời kết nối dữ liệu thành một cơ sở dữ liệu thống nhất.



Hình 14. Sơ đồ kho lưu trữ dữ liệu

Các vấn đề của dữ liệu cần tích hợp

1. **Vấn đề nhận dạng thực thể - Entity identification problem**

   Đây là vấn đề trùng lặp dữ liệu giữa **nhiều** trường dữ liệu trong **nhiều** cơ sở dữ liệu khác nhau.

   Tích hợp dữ liệu là quá trình thu thập thông tìn từ nhiều nguồn khác nhau thành một kho dữ liệu bao gồm nhiều cơ sở dữ liệu, tệp dữ liệu, loại dữ liệu,... Khi đó, khó tránh khỏi việc trùng lặp dữ liệu ở các cở sở dữ liệu khác nhau. Vậy làm thế nào để có thể ánh xạ tất cả các thông tin liên quan tới một thực thể (entity) từ nhiều nguồn dữ liệu khác nhau thành một trong kho dữ liệu cuối cùng?

Ví dụ: làm sao để ánh xạ customer_id của dữ liệu này với cust_number của dữ liệu khác?

2. **Dữ liệu thừa và phân tích tương quan**

Đây là vấn đề trùng lắp dữ liệu giữa **nhiều** trường dữ liệu trong **cùng một** cơ sở dữ liệu.

Dư thừa dữ liệu là một trong những vấn đề quan trọng cần được giải quyết trong tích hợp dữ liệu. Dữ liệu thừa là những dữ liệu cùng thể hiện một thông tin nhưng thể hiện dưới các thuộc tính khác nhau.

Ví dụ: Ta có bảng dữ liệu dưới đây

- *Is_veg: 1 nếu pizza chay, 0 nếu ngược lại*
- *Is_nonveg: 1 nếu pizza không chay, 0 nếu ngược lại*

Pizza_name	Is_veg	Is_nonveg
Farm House	1	0
Veg Loaded	1	0
Chicken Sausage	0	1
Non-Veg Supreme	0	1
Chicken Fiesta	0	1
Veg Extravaganza	1	0
Deluxe Veggie	1	0

Ta có thể thấy rằng, bảng dữ liệu tiêu tốn hai thuộc tính Is_veg và Is_nonveg chỉ để thể hiện chiếc "pizza" có phải đồ chay hay không trong khi chỉ cần một là đủ. Đây chính là dữ liệu thừa.

3. **Chuỗi giá trị lặp lại**

Đây là vấn đề trùng lắp dữ liệu trong **cùng một** trường dữ liệu trong **cùng một** cơ sở dữ liệu.

Ngoài xử lý vấn đề dư thừa dữ liệu ở cấp thuộc tính dữ liệu, tích hợp dữ liệu còn có các vấn đề ở cấp chuỗi giá trị, như có nhiều hơn hai chuỗi dữ liệu khi thuộc tính đã được đảm bảo tính đặc trưng. Mâu thuẫn trong thu

thập dữ liệu thường xảy ra, do cách thiết lập thông tin đầu vào không hợp lý, hoặc việc cập nhật thông tin không đồng bộ.

Ví dụ, khi cơ sở dữ liệu cập nhật thông tin mua hàng của một khách hàng, thông tin đầu vào thay vì khóa chính lại là tên và địa chỉ mua hàng, chênh lệch thông tin sẽ xảy ra, khi đó một cái tên sẽ xuất hiện trong cở sở dữ liệu nhiều lần dưới nhiều địa chỉ khác nhau.

4. **Mâu thuẫn trong định nghĩa dữ liệu**

Đây là vấn đề ngầm trùng lắp dữ liệu do khác định nghĩa và đơn vị giữa các trường.

Tích hợp dữ liệu cũng bao gồm xác định và giải quyết mâu thuẫn trong dữ liệu. Trong thế giới thực, giá trị của các thuộc tính trong bộ dữ liệu có thể được quy định, mô tả, lưu trữ theo những cách khác nhau. Ví dụ, thuộc tính cân nặng *weights* có thể được lưu với hệ đo lường khối lượng của Mỹ (pound) ở nguồn này, nhưng cũng có thể được lưu với hệ đo lường khối lượng của Việt Nam (kilogram) ở nguồn khác. Hoặc như tiêu chuẩn đánh giá học sinh giữa các trường, ở trường này học sinh sẽ được đánh giá theo thang điểm từ F tới A, còn ở trường khác sẽ được đánh giá theo thang điểm từ 1 tới 10.

Mâu thuẫn cũng có thể xảy ra do các thuộc tính đặt tên trùng nhau nhưng lại lưu trữ thông tin ở các mức độ khác nhau. Ví dụ như thuộc tính $total\_sales$ ở cơ sở dữ liệu này thể hiện doanh số của một công ty con thuộc công ty *AllElectronics*, nhưng ở cơ sở dữ liệu khác thuộc tính $total\_sales$ lại là doanh số của một trong những cửa hàng của *AllElectronics*.

2.3 Các phương pháp tích hợp dữ liệu

2.3.1 Vấn đề nhận dạng thực thể - Entity identification problem

Ta có thể gộp các cở sở dữ liệu về thành một cơ sở dữ liệu thống nhất, và đặc biệt chú ý tới việc quy định cấu trúc của dữ liệu. Việc này cũng nhằm đảm

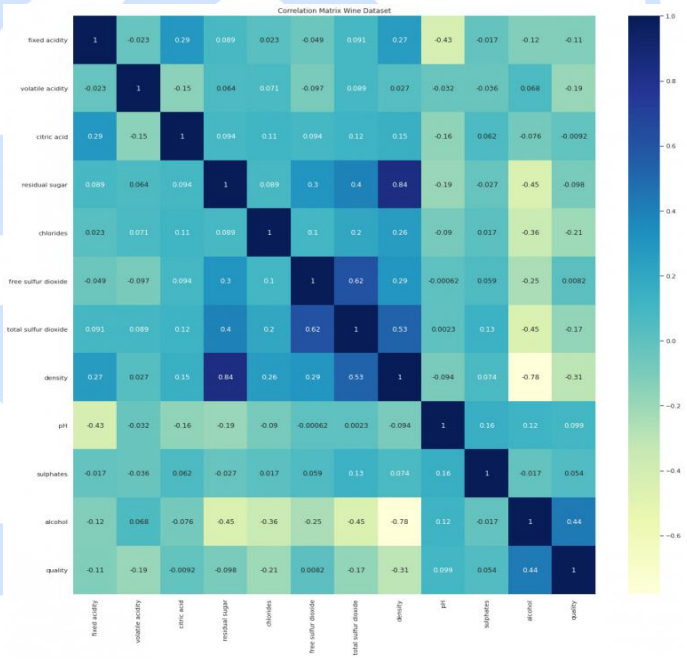
bảo sự tương quan giữa các thuộc tính giữa các cơ sở dữ liệu hay khóa ngoại giữa cơ sở dữ liệu gốc tới cơ sở dữ liệu đích.

2.3.2 Dữ liệu thừa và phân tích tương quan

Ta có thể nhìn vấn đề này rằng trong một cơ sở dữ liệu có 2 hay nhiều dữ liệu có tương quan cao với nhau. Thay đổi trường dữ liệu này sẽ thay đổi trực tiếp đến trường dữ liệu kia và ngược lại. Hay nói cách khác, chúng có thể biểu diễn lẫn nhau.

Việc chỉ giữ lại một trong số chúng và loại bỏ các trường dữ liệu còn lại là việc cần thiết để giảm thời gian tính toán và khối lượng lưu trữ

Trong mức độ bài tập này, chúng tôi chỉ đề ra phương pháp tính mức độ tương quan giữa các trường dữ liệu. Còn việc chọn lựa trường dữ liệu cần thiết để loại bỏ còn phụ thuộc vào các yếu tố dữ liệu khác.



Hình 15. Ma trận tương quan các đặc trưng dữ liệu

Các phương pháp tính độ tương quan

1. *Kiểm tra tương quan X^2 cho dữ liệu định danh*

Kiểm tra tương quan X^2 (chi-square) là phương pháp so sánh giữa tần số kỳ vọng (nghĩ/giả thuyết) và tần số quan sát (thấy/thực tế) dùng để xác định sự độc lập giữa các thuộc tính trong khai thác dữ liệu.

Công thức đó được triển khai trên dữ liệu định danh như sau, giả thiết ta có hai thuộc tính A và B, A có c giá trị riêng biệt là $a_1, a_2, \dots, a_c$ và B có r giá trị riêng biệt là $b_1, b_2, \dots, b_r$. Tạo bảng tương quan giữa hai thuộc tính A và B với A là cột và B là dòng, ta sẽ có một bảng tương quan giá trị với kích thước $c * r$. Mỗi dữ liệu (A_i, B_j) đều có ô và giá trị riêng. Giá trị X^2 sẽ được tính như sau:

$$\mathrm{x}^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}},$$

với o_{ij} là tần số quan sát, Sum để công thức có thể duyệt qua toàn bộ các ô dữ liệu trong bảng và e_{ij} là tần số kỳ vọng có thể tính toán như sau:

$$e_{ij} = \frac{count(A = a_i) * count(B = b_j)}{n},$$

với n là số chuỗi dữ liệu (data tuples), $count(A = a_i)$ là tổng số lần xuất hiện của giá trị a_i trong thuộc tính A, còn $count(B = b_j)$ là tổng số lần xuất hiện giá trị b_j trong thuộc tính B.

Tiền đề của công thức X^2 là A và B độc lập với nhau, tức là công thức ban đầu không xét tới bất cứ mối tương quan nào giữ hai thuộc tính này. X^2 càng lớn thì tức là sự phụ thuộc lẫn nhau giữa hai thuộc tính càng lớn, ngược lại, nếu x^2 là $Null$, vậy tức là hai thuộc tính A và B độc lập với nhau và không có mối tương quan nào.

Ví dụ: Một khảo sát thực hiện trên 1500 người với thông tin bao gồm giới tính và sở thích đọc của người đó là tiểu thuyết hay không phải tiểu thuyết. Áp dụng công thức (2.1) ta có tương quan giữa giới tính *nam* và sở thích đọc *tiểu thuyết* là:

$$e_{11} = \frac{count(male) * count(fiction)}{n} = \frac{300 * 450}{1500} = 90$$

ta có 90 ở đây là tần số kỳ vọng, là giá trị dự đoán (giá trị trong dấu ngoặc đơn bên dưới) ta kỳ vọng rằng sẽ có bấy nhiêu nam giới thích đọc tiểu thuyết, ta tiếp tục áp dụng công thức này cho các mối liên hệ $male\ x\ non\_fiction$, $female\ x\ fiction$ và $female\ x\ non\_fiction$.

Sau khi thu thập được dữ liệu thực tế, ta áp dụng công thức (2.1) để tính toán sự chênh lệch giữa hai loại giữ liệu cho từng ô trong bảng tương quan và tiếp tục như vậy cho đến hết giá trị của bảng.

Lưu ý rằng tổng các giá trị thực tế và tổng giá trị dự đoán phải bằng tổng giá trị sau cùng.

Bảng 3. Khảo sát sở thích đọc sách ở nam và nữ

	male	*female*	*Total*
fiction	250 (90)	200 (360)	450
non_fiction	50 (210)	1000 (840)	1050
Total	300	1200	1500

Tương tự như trên, ta có:

$$x^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$
$$= 288.44 + 121.90 + 71.11 + 30.48 = 507.93$$

Do giá trị x^2 ta tính toán được bằng 507.93, giá trị khá lớn có nghĩa là A và B phụ thuộc lên nhau và tương quan lẫn nhau.

2. Hệ số tương quan cho dữ liệu kiểu số

Với thuộc tính dữ liệu dạng số, ta có thể dễ dàng tính toán độ tương quan giữa hai thuộc tính A và B bằng cách tính hệ số tương quan *Pearson (Pearson's product moment coefficient).*

$$r_{A,B} = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n\sigma_A\sigma_B} = \frac{\sum_{i=1}^{n}(a_i b_i) - (n\overline{AB})}{n\sigma_A\sigma_B},$$

với n là số chuỗi dữ liệu, a_i, b_i là giá trị của A và B trong chuỗi dữ liệu i, $\bar{A}$ và $\bar{B}$ là giá trị trung bình của A và B, σ_A và σ_B là độ lệch chuẩn của A và B. Lưu ý rằng $-1 \leq r_{A,B} \leq +1$. Nếu $r_{A,B}$ lớn hơn 0, tức là A, B tương quan và A, B tỉ lệ thuận với nhau, giá trị càng lớn, độ tương quan càng cao.

Nếu $r_{A,B}$ bằng 0, tức là giữa A và B không có mối tương quan nào. Bé hơn 0 tức là giữa A và B là mối quan hệ tương quan nghịch, A tăng khi B giảm và ngược lại.

Cần lưu ý rằng mối tương quan không phải quan hệ nhân quả. Dù A tương quan với A, B cũng không có nghĩa rằng A gây ra B và ngược lại. Ví dụ, khi phân tích số liệu ở một thị trấn, người ta phát hiện ra rằng số lượng bệnh viện tương quan với số vụ trộm mèo trong thị trấn, điều đó không có nghĩa là xây thêm bệnh viện sẽ khiến mèo bị trộm nhiều hơn, mà cả 2 yếu tố trên chỉ tương quan với nhau thông qua *dân số*.

3. Hiệp phương sai của dữ liệu kiểu số

Trong xác suất thống kê, tương quan và hiệp phương sai là hai phép tính thường dùng để tính toán sự ảnh hưởng lẫn nhau của dữ liệu. Giả dụ ta có hai thuộc tính A, B và chuỗi các giá trị (a_1, b_1), … (a_n, b_n). Giá trị trung bình của A và B được tính như sau:

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n}(a_i)}{n}$$

và

$$E(B) = \bar{B} = \frac{\sum_{i=1}^{n}(b_i)}{n}.$$

Hiệp phương sai của A và B được tính như sau:

$$Cov(A, B) = E\big((A - \bar{A})(B - \bar{B})\big) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})(b_i - \bar{B})}{n},$$

So sánh giữa công thức (2.3) và (2.4), tương quan và hiệp phương sai, ta thấy rằng,

$$r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B},$$

với σ_A và σ_B là độ lệch chuẩn của A và B, công thức trên có thể diễn giải như sau:

$$Cov(A,B) = E(A * B) - \overline{AB}$$

Khi A và B tương quan, nếu A lớn hơn giá trị trung bình $\bar{A}$, thì B cũng sẽ hơn giá trị trung bình $\bar{B}$, do đó hiệp phương sai của A và B luôn dương. Mặt khác, nếu một trong hai thuộc tính bé hơn giá trị trung bình của nó và thuộc tính còn lại thì lớn hơn, hiệp phương sai của chúng sẽ âm.

Khi A và B độc lập với nhau, $E(A * B) = E(A) * E(B)$. Suy ra, hiệp phương sai $Cov(A,B) = E(A * B) - \overline{AB} = E(A) * E(B) - \overline{AB}$.. Tuy nhiên, điều ngược lại thì không đúng. Nếu hai thuộc tính độc lập với nhau, hiệp phương sai giữa chúng sẽ bằng 0, nhưng hiệp phương sai bằng 0 không có nghĩa là hai thuộc tính độc lập với nhau.

Bảng 4. Tỷ giá cổ phiếu của AllElectronics và HighTech

Time point	*AllElectronics*	*HighTech*
t1	6	20
t2	5	10
t3	4	14
t4	3	5
t5	2	5

Ví dụ: Tính hiệp phương sai của bảng giá trị trên. Ở bảng trên, ta có giá cổ phiếu của hai công ty *AllElectroics* và *HighTech*, giả thiết thị trường xảy ra khủng hoảng làm ảnh hưởng đến giá trị cổ phiếu, liệu chúng có cùng giảm?

$$E(AllElectronics) = \frac{6+5+4+3+2}{5} = \frac{20}{5} = \$4$$

và

$$E(HighTech) = \frac{20 + 10 + 14 + 5 + 5}{5} = \frac{54}{5} = \$10.80$$

Dùng công thức (2.4), ta có hiệp phương sai:

$$Cov(AllElectronics, HighTech)$$
$$= \frac{6 * 20 + 5 * 10 + 4 * 14 + 3 * 5 + 2 * 5}{5} - 4 * 10.80$$
$$= 50.2 - 43.2 = 7$$

Từ giá trị hiệp phương sai, ta thấy rằng tỷ giá cổ phiếu của hai công ty tương quan và cùng đi lên.

2.3.3 Chuỗi giá trị lặp lại

Tối ưu hóa quá trình thu thập dữ liệu bằng cách thiết lập thông tin đầu vào một cách hợp lý và đồng bộ hoá quá trình cập nhật thông tin.

Cụ thể trong ví dụ khách hàng nêu trên, ta có thể áp dụng phương pháp chuẩn hóa dữ liệu:

1. Đầu tiên, ta cần tạo một bảng mới chứa thông tin khách hàng, bao gồm tên và địa chỉ.
2. Sau đó, ta tạo một bảng khác để lưu thông tin mua hàng, bao gồm khóa chính của mỗi hóa đơn và khóa ngoại của bảng khách hàng.
3. Khi có thông tin mua hàng mới, ta sẽ kiểm tra xem khách hàng đã tồn tại trong bảng khách hàng hay chưa.
4. Nếu chưa, ta thêm mới khách hàng và lấy khóa chính của khách hàng để làm khóa ngoại cho bảng mua hàng.
5. Nếu đã tồn tại, ta sẽ lấy khóa chính của khách hàng đó để làm khóa ngoại cho bảng mua hàng.

Với cách xử lý này, ta sẽ giảm được dư thừa dữ liệu trong cơ sở dữ liệu và đảm bảo tính nhất quán của dữ liệu.

2.3.4 Xác định mâu thuẫn dữ liệu và phân giải

Như ta đã thấy, vấn đề mâu thuẫn dữ liệu do dữ liệu được lưu trữ dưới các thang đo khác nhau hay được định nghĩa khác nhau thì không thể giải quyết bằng các kỹ thuật xử lý dữ liệu như trên.

Đối với trường hợp này chỉ có thể giải quyết bằng phương pháp thủ công, chính là người quản lý dữ liệu tự định nghĩa lại và quy ước dữ liệu về một thể thống nhất bằng kiến thức của mình.

III. Tham khảo

I. Tham khảo

[1] [Online]. Available: https://web.cs.hacettepe.edu.tr/~ilyas/Courses/VBM684/lec03-DataPreprocessing.pdf.

[2] [Online]. Available: http://hanj.cs.illinois.edu/cs412/bk3/03.pdf.