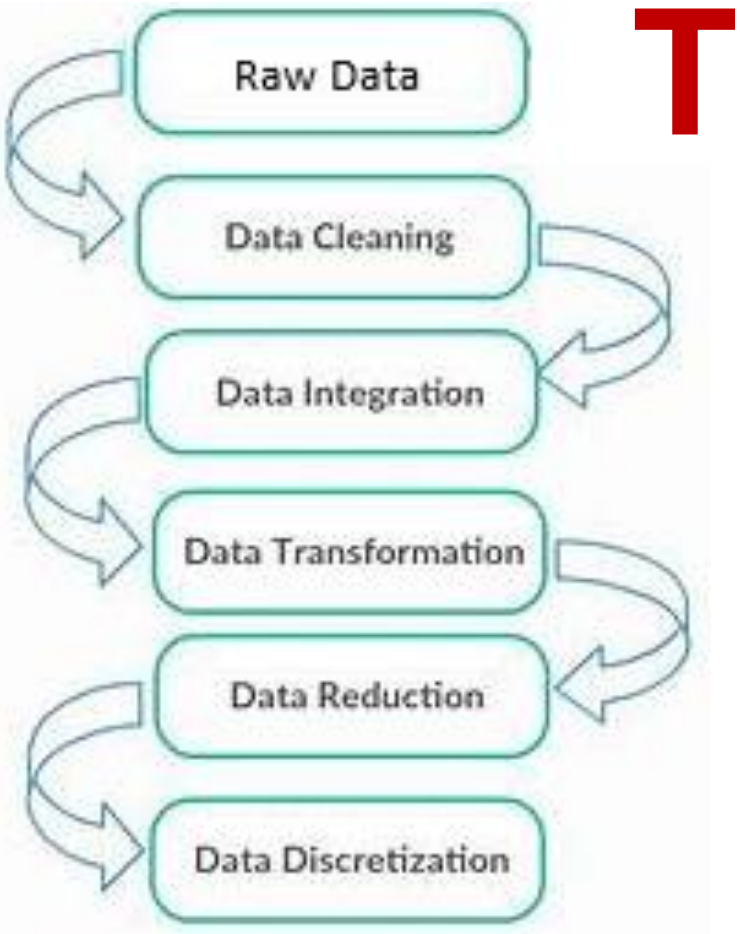
KHAI THÁC DỮ LIỆU
(Data Mining)

Lê Văn Hạnh
levanhanhvn@gmail.com

NỘI DUNG MÔN HỌC

1. Tổng quan về Data Science
2. Tìm hiểu dữ liệu
3. Tiền xử lý dữ liệu
4. Khai thác các mẫu phổ biến, mối kết hợp và mối tương quan
5. Phân loại (*Classification*)
6. Phân tích cụm (*Cluster analysis*)

Chương 3



TIỀN XỬ LÝ DỮ LIỆU

(Data Preprocessing)

Lê Văn Hạnh

levanhanhvn@gmail.com

NỘI DUNG CHƯƠNG 3

1. Tổng quan về xử lý dữ liệu (*Data Preprocessing: An Overview*)
2. Làm sạch dữ liệu (*Data Cleaning*)
3. Tích hợp dữ liệu (*Data Integration*)
4. Giảm thiểu dữ liệu (*Data Reduction*)
5. Chuyển đổi dữ liệu và phân tách dữ liệu (*Data Transformation and Data Discretization*)
6. Thực hành tiền xử lý dữ liệu

1. TỔNG QUAN VỀ XỬ LÝ DỮ LIỆU (*Data Preprocessing: An Overview*)

- Dữ liệu phát sinh trong quá trình tác nghiệp được gọi là dữ liệu thô (raw/original data).
- Cơ sở dữ liệu trong thế giới thực thường có:
  - Kích thước lớn (thường là vài gigabyte trở lên)
  - Nguồn gốc từ nhiều nguồn không đồng nhất (*heterogenous sources*).
  - Không hoàn chỉnh: thiếu thuộc tính, thiếu giá trị cần dung
  - Chứa giá trị nhiễu: có lỗi hoặc có giá trị lệch.
  - Không nhất quán

1. Tổng quan về xử lý dữ liệu (*Data Preprocessing: An Overview*)

- Do đó CSDL rất dễ bị:
  - Nhiễu (*noisy*)
  - Thiếu dữ liệu (*missing data*)
  - Không nhất quán (*inconsistent*)

    $\Rightarrow$ chất lượng dữ liệu thấp.
- Dữ liệu chất lượng thấp sẽ dẫn đến kết quả khai thác chất lượng thấp $\Rightarrow$ dữ liệu cần được xử lý trước để giúp cải thiện chất lượng dữ liệu và do đó, cải thiện kết quả khai thác.
- Cho dù chất lượng dữ liệu không thấp, nhưng để khai thác các khía cạnh của chúng, thường phải được biến đổi về dạng/kiểu dữ liệu thích hợp

1.1.- Các yếu tố ảnh hưởng đến chất lượng dữ liệu

- Các yếu tố tạo nên chất lượng dữ liệu:
  - Độ chính xác (*accuracy*)
  - Tính đầy đủ (*completeness*)
  - Tính nhất quán (*consistency*)
  - Tính kịp thời
  - Độ tin cậy
  - Khả năng diễn giải

Các yếu tố chính xác định chất lượng dữ liệu

- Lưu ý: chất lượng dữ liệu phụ thuộc vào mục đích sử dụng dữ liệu. Hai người dùng khác nhau có thể có những đánh giá rất khác nhau về chất lượng của một CSDL nhất định.

1.1.- Các yếu tố ảnh hưởng đến chất lượng dữ liệu

- Các yếu tố chính xác định nên chất lượng dữ liệu:
  - ***Dữ liệu không chính xác*** (*inaccuracy*): Có nhiều lý do có thể dẫn đến dữ liệu không chính xác như:
    - *Lỗi khi thu thập dữ liệu*: Người dùng có thể cố tình gửi giá trị dữ liệu không chính xác khi họ không muốn gửi thông tin cá nhân (ví dụ: chọn giá trị mặc định "Ngày 1/1" cho ngày sinh nhật). Điều này được gọi là dữ liệu bị thiếu được ngụy trang.
    - *Lỗi trong quá trình truyền dữ liệu*.
    - *Lỗi có thể do những hạn chế về công nghệ*: như kích thước bộ đệm hạn chế để điều phối việc truyền và tiêu thụ dữ liệu được đồng bộ hóa.
    - *Lỗi do sự không nhất quán trong quy ước*: đặt tên hoặc mã dữ liệu hoặc định dạng không nhất quán cho các trường đầu vào (ví dụ: ngày).
    - *Lỗi do dữ liệu trùng lặp*.

1.1.- Các yếu tố ảnh hưởng đến chất lượng dữ liệu

- Các yếu tố chính xác định nên chất lượng dữ liệu:
  - ***Dữ liệu không đầy đủ*** (*incomplete data*): Các thuộc tính quan tâm có thể không phải lúc nào cũng có sẵn, chẳng hạn như:
    - Vì chúng không được coi là quan trọng tại thời điểm nhập dữ liệu.
    - Dữ liệu liên quan có thể không được ghi lại do hiểu lầm hoặc do trục trặc của thiết bị.
  - ***Dữ liệu không nhất quán*** (*inconsistency*): chủ yếu là do chưa hoặc không có quy định thống nhất chung khi nhập dữ liệu như tên gọi cho đối tượng (sản phẩm, tên địa danh, …), đơn vị tính, …

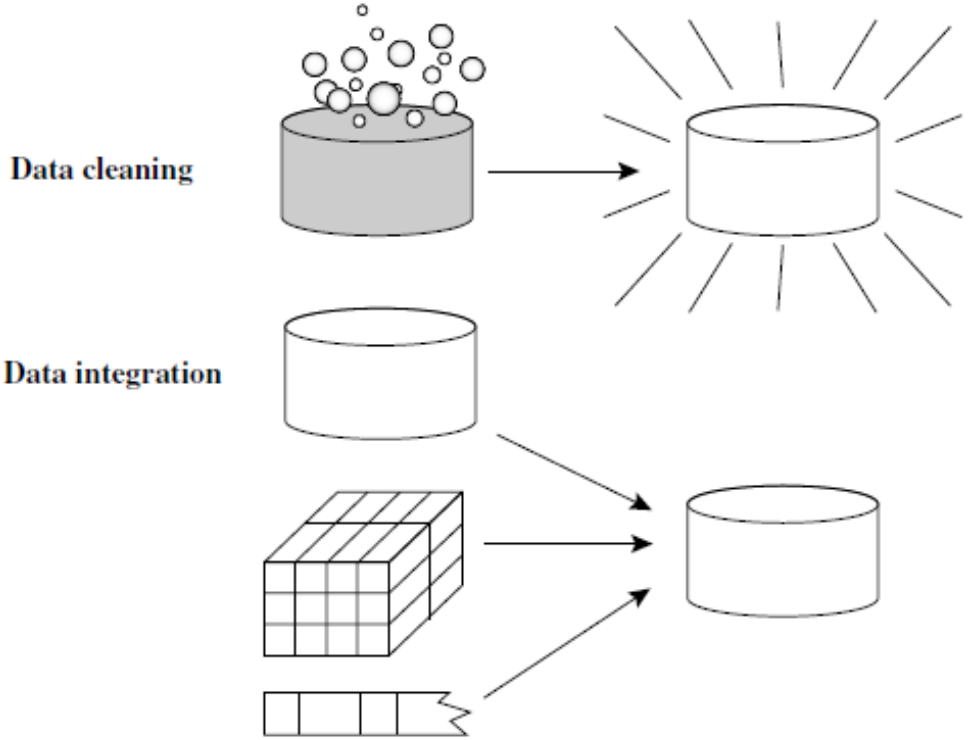
1.1.- Các yếu tố ảnh hưởng đến chất lượng dữ liệu

- Các yếu tố khác xác định nên chất lượng dữ liệu:
  - ***Tính kịp thời*** (*timeliness*): Giả sử cần giám sát việc phân phối tiền thưởng về doanh số bán hàng hàng tháng cho các nhân viên. Tuy nhiên, một số khách hàng của nhân viên không thanh toán tiền hàng với nhân viên đúng hạn vào cuối tháng dẫn đến việc cập nhật dữ liệu cuối tháng không kịp thời cũng đã ảnh hưởng tiêu cực đến chất lượng dữ liệu.
  - ***Độ tin cậy*** (*believability*): phản ánh mức độ tin cậy của dữ liệu đối với người dùng. Giả sử rằng một cơ sở dữ liệu tại một thời điểm có một số lỗi nhưng tất cả các lỗi đó đều đã được sửa chữa. Tuy nhiên, những lỗi trong quá khứ đã gây ra nhiều vấn đề cho người dùng bộ phận bán hàng và khiến họ không còn tin tưởng vào dữ liệu nữa.
  - ***Khả năng diễn giải*** (*interpretability*): phản ánh mức độ dễ hiểu của dữ liệu. Giả sử dữ liệu sử dụng nhiều mã số kế toán mà bộ phận kinh doanh không biết diễn giải. Mặc dù cơ sở dữ liệu hiện tại chính xác, đầy đủ, nhất quán và kịp thời nhưng người dùng của bộ phận bán hàng có thể coi nó là chất lượng thấp do độ tin cậy và khả năng diễn giải kém.

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

Tiền xử lý dữ liệu

- Làm sạch dữ liệu *(Data Cleaning)*
- Tích hợp dữ liệu *(Data Integration)*
- Giảm thiểu dữ liệu *(Data Reduction)*
- Chuyển đổi dữ liệu *(Data Transformation)*



1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

i. Xử lý dữ liệu bị thiếu (*missing data*)

- Là dữ liệu không có sẵn khi cần được dung.
- Nguyên nhân:
  - Khách quan: không tồn tại lúc nhập liệu, sự cố, …
  - Chủ quan: tác nhân con người

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

i. Xử lý dữ liệu bị thiếu (*missing data*)

- Các giải pháp xử lý khi dữ liệu bị thiếu:
  a. Bỏ qua bộ dữ liệu
  b. Điền giá trị còn thiếu theo cách thủ công
  c. Sử dụng hằng số chung (thông quá hằng số toàn cục, giá trị dự đoán, …) để điền giá trị còn thiếu
  d. Sử dụng thước đo xu hướng trung tâm (giá trị phổ biến nhất, trung bình toàn cục) cho thuộc tính.
  e. Sử dụng thuộc tính trung bình hoặc trung vị cho tất cả các mẫu thuộc cùng loại với bộ dữ liệu đã cho
  f. Sử dụng giá trị có xác suất lớn nhất để điền vào giá trị còn thiếu

Nhận xét:

- Phương pháp từ (c) đến (f) làm sai lệch dữ liệu – do giá trị được điền vào có thể không chính xác.
- Tuy nhiên, Phương pháp (f) là một chiến lược phổ biến.
- Ngăn chặn dữ liệu bị thiếu: thiết kế tốt CSDL và các thủ tục nhập liệu (các ràng buộc dữ liệu)

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

ii. Nhận diện phần tử biên (*outliers*)

- Là những dữ liệu (đối tượng) không tuân theo đặc tính/hành vi chung của tập dữ liệu (đối tượng).
- Giải pháp nhận diện phần tử biên:
  - Dựa trên phân bố thống kê (*statistical distribution-based*)
  - Dựa trên khoảng cách (*distance-based*)
  - Dựa trên mật độ (*density-based*)
  - Dựa trên độ lệch (*deviation-based*)

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iii.Giảm thiểu dữ liệu nhiễu (*noisy data*)

- Là những dữ liệu ngoại lệ (*exceptions*) hoặc là sai số hoặc phương sai ngẫu nhiên của một biến.
- Các phương pháp trực quan hóa dữ liệu (được giới thiệu trong chương trước (biểu đồ hình hộp - *box plots* - và biểu đồ phân tán - *scatter plots*) có thể được sử dụng để xác định các ngoại lệ, có thể biểu thị nhiễu.

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iii.Giảm thiểu dữ liệu nhiễu (*noisy data*)

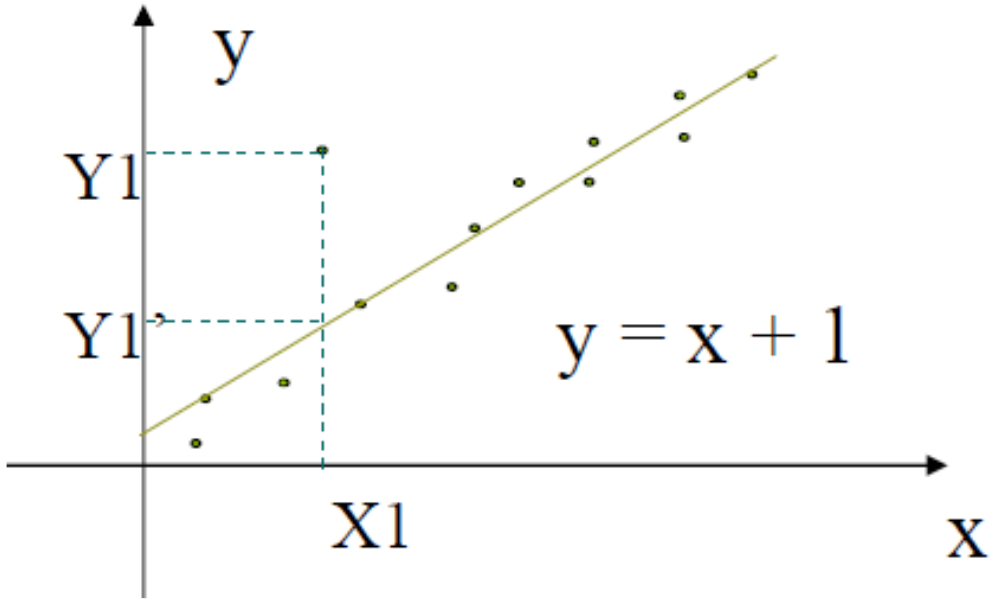
- Các kỹ thuật làm mịn (*smooth*) dữ liệu
  - Hồi quy (*regression*)
  - Phân tích cụm (*cluster*)
  - Tạo nhóm *(Binning techniques)*
  - Hệ thống phân cấp khái niệm (*Concept hierarchies*)

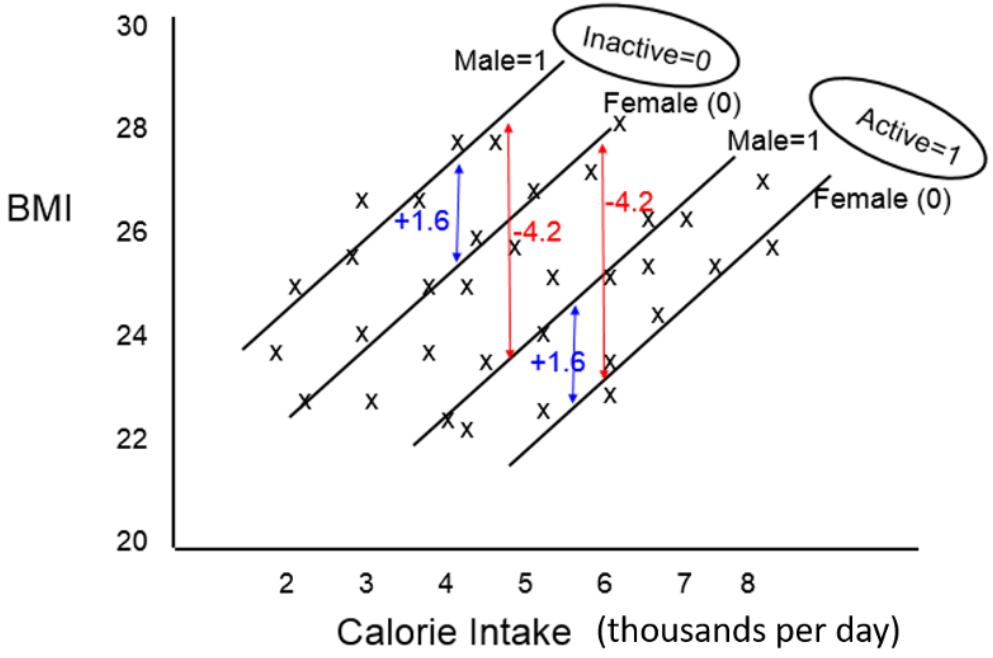
1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iii.Giảm thiểu dữ liệu nhiễu (*noisy data*) **bằng hồi quy** *(regression)*

- Hồi quy tuyến tính (*linear regression*) liên quan đến việc tìm ra đường "tốt nhất" phù hợp với ***hai*** thuộc tính (hoặc biến) để một thuộc tính có thể được sử dụng để dự đoán thuộc tính kia.
- Hồi quy tuyến tính bội (*multiple linear regression*) là phần mở rộng của hồi quy tuyến tính, trong đó có ***nhiều hơn hai*** thuộc tính có liên quan và dữ liệu phù hợp với bề mặt đa chiều.



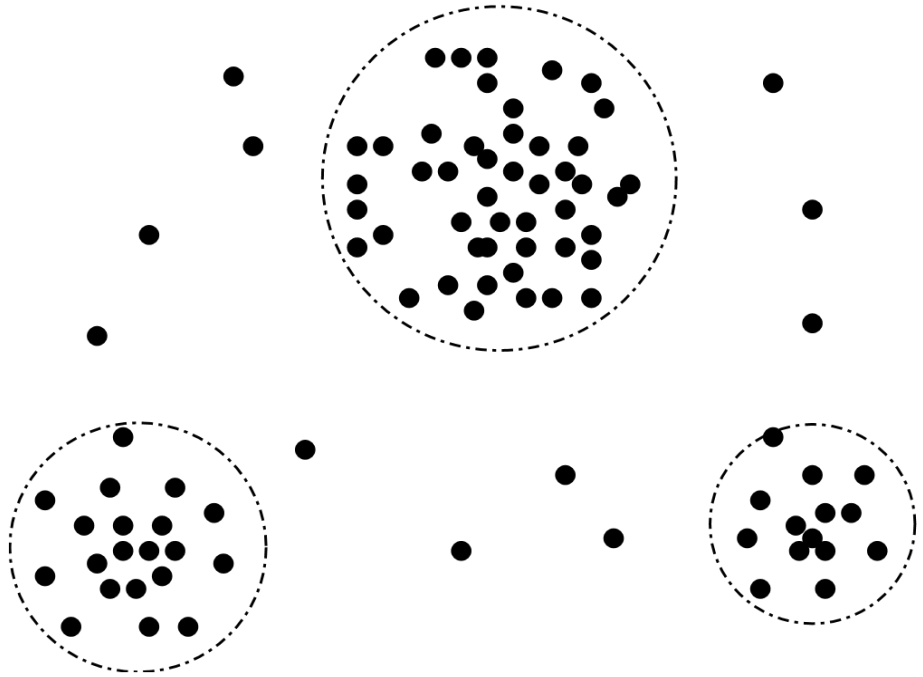


1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iii.Giảm thiểu dữ liệu nhiễu (*noisy data*) **bằng phân tích ngoại lệ** *(Outlier analysis)*

- Các ngoại lệ có thể được phát hiện bằng cách phân cụm (*clustering*), ví dụ: trong đó các giá trị tương tự được tổ chức thành các nhóm (*groups*) hoặc "cụm" (*clusters*).
- Ví dụ: theo trực giác, các giá trị nằm ngoài tập hợp các cụm có thể được coi là các giá trị ngoại lệ.



1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iii.Giảm thiểu dữ liệu nhiễu (*noisy data*) bằng tạo nhóm *(Binning techniques)*

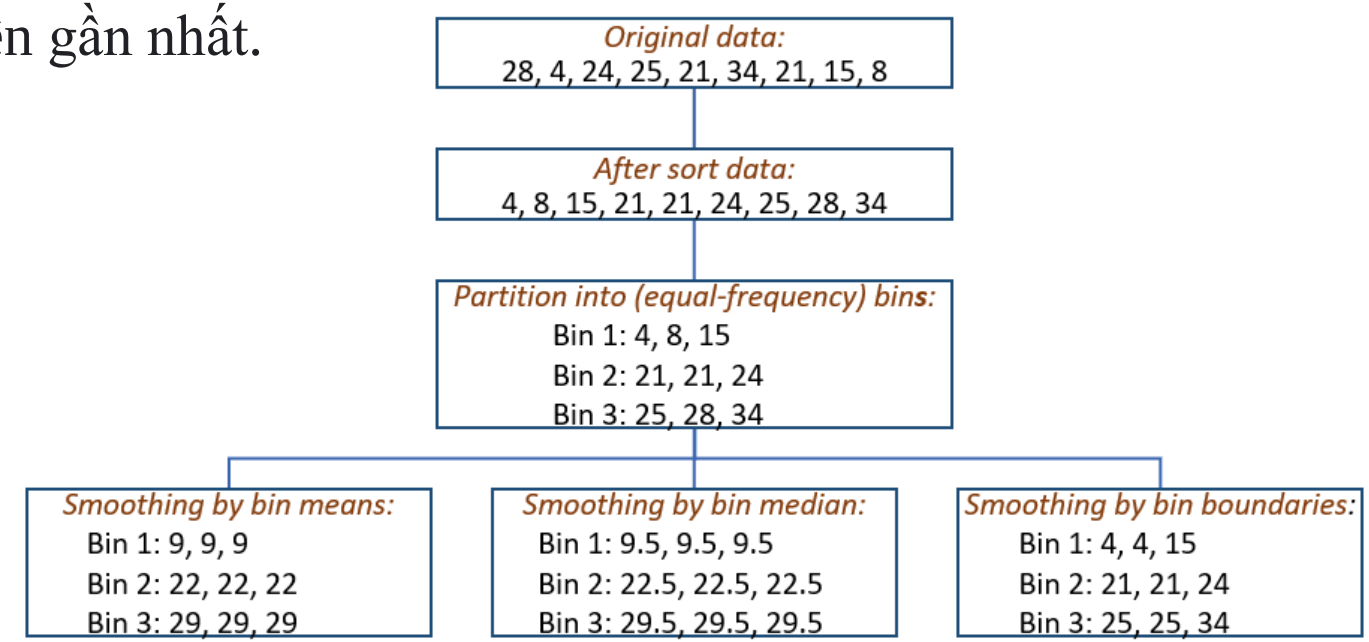
- Giúp giảm số lượng giá trị riêng biệt cho mỗi thuộc tính. Điều này hoạt động như một hình thức rút gọn dữ liệu cho các phương pháp khai thác dữ liệu dựa trên logic (*logic-based data mining methods*), chẳng hạn như quy nạp cây quyết định (*decision tree induction*), liên tục thực hiện so sánh giá trị trên dữ liệu đã được sắp xếp.
- Các phương pháp liên kết làm mịn một giá trị dữ liệu đã được sắp xếp bằng cách tham khảo các giá trị "vùng lân cận" ("neighborhood") của nó.
- Các giá trị đã sắp xếp được phân phối vào một số "nhóm" (buckets) hoặc thùng (bins).
- Sau đó thực hiện làm mịn cục bộ (local smoothing).

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iii.Giảm thiểu dữ liệu nhiễu (*noisy data*) **bằng tạo nhóm** *(Binning techniques)*

- Ví dụ: Cho dữ liệu gồm: 28, 4, 24, 25, 21, 34, 21, 15, 8.
  - Chia dữ liệu vào các bin có số lượng (hay tần suất) bằng nhau (3 giá trị/ngăn).
  - Thay thế giá trị trong bin bằng 1 trong 3 cách sau:
    - Dùng *mean*: Thay thế các giá trị trong bin = giá trị trung bình của bin.
    - Dùng *median*: Thay thế các giá trị trong bin = giá trị trung vị của bin.
    - Dùng *smoothing by bin boundaries* (làm mịn theo ranh giới của bin): giá trị tối thiểu và tối đa trong một bin nhất định được xác định là ranh giới của bin. Mỗi giá trị còn lại trong bin được thay thế bằng giá trị biên gần nhất.

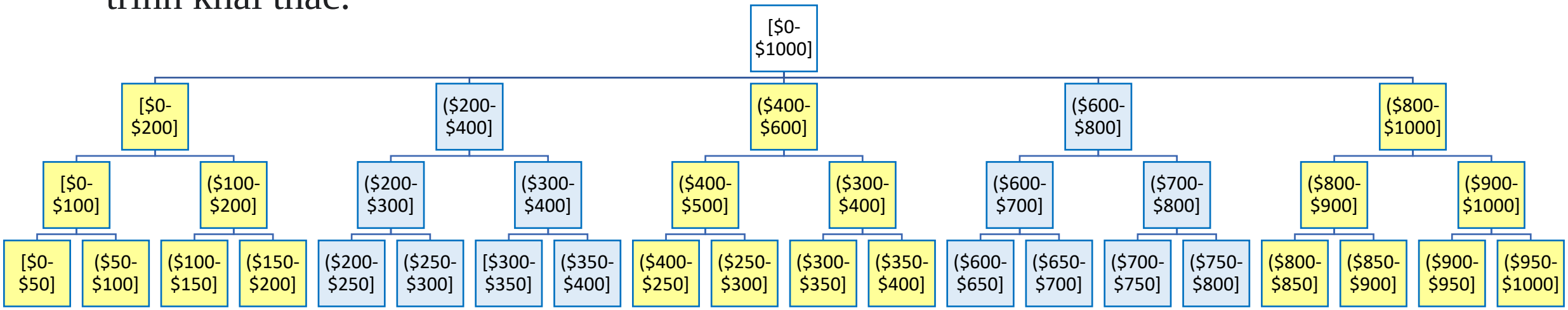


1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iv.Giảm thiểu dữ liệu nhiễu (*noisy data*) bằng *Hệ thống phân cấp khái niệm (Concept hierarchies)*

- Là một dạng rời rạc hóa dữ liệu cũng có thể được sử dụng để làm mịn dữ liệu.
- Ví dụ: hệ thống phân cấp khái niệm về giá có thể ánh xạ các giá trị giá thực thành giá rẻ tiền, giá vừa phải và đắt, do đó làm giảm số lượng giá trị dữ liệu được xử lý trong quá trình khai thác.

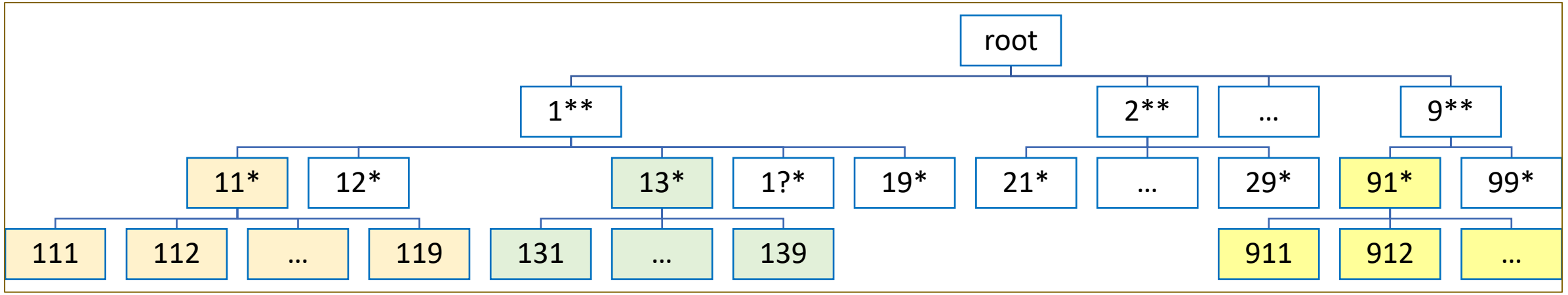


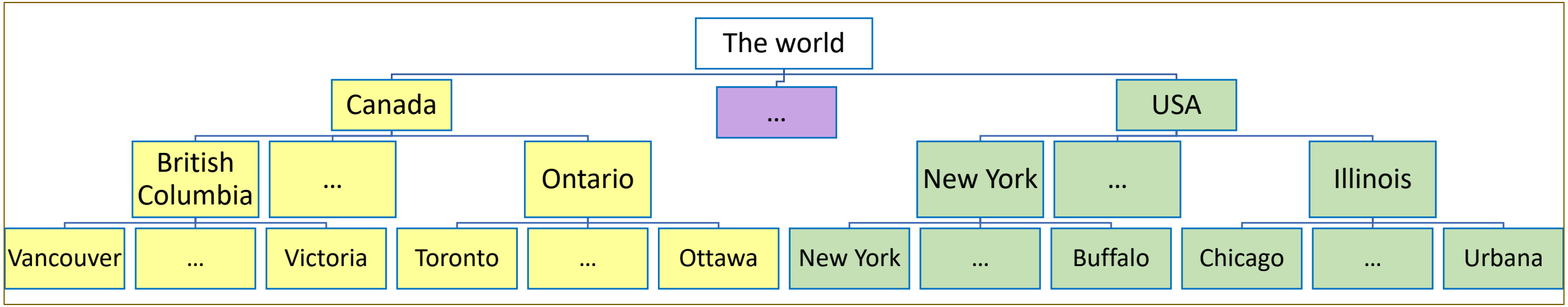
1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

iv. Giảm thiểu dữ liệu nhiễu (*noisy data*) **bằng *Hệ thống phân cấp khái niệm*** (*Concept hierarchies*)

– Minh họa một số *Hệ thống phân cấp khái niệm*





1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

- Làm sạch dữ liệu gồm:

iv.Xử lý dữ liệu không nhất quán *(inconsistent data)*

- Là những dữ liệu (đối tượng) không tuân theo đặc tính/hành vi chung của tập dữ liệu (đối tượng).
- Giải pháp xử lý dữ liệu không nhất quán
  - Tận dụng siêu dữ liệu, rang buộc dữ liệu, sự kiểm tra của nhà phân tích dữ liệu cho việc nhận diện
  - Điều chỉnh dữ liệu không nhất quán bằng tay.
  - Biến đổi, chuẩn hóa dữ liệu tự động.

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.1. Làm sạch dữ liệu (Data cleaning)

- Nếu người dùng tin rằng dữ liệu bị bẩn (dirty), họ **khó có thể tin tưởng vào kết quả** của bất kỳ hoạt động **khai thác dữ liệu** nào đã được áp dụng.
- Dữ liệu bẩn có thể gây nhầm lẫn cho quy trình khai thác, dẫn đến **kết quả đầu ra không đáng tin cậy**.
- Mặc dù hầu hết các quy trình khai thác đều có một số quy trình để xử lý dữ liệu không đầy đủ hoặc nhiễu nhưng chúng không phải lúc nào cũng hiệu quả.
- Do đó, bước tiền xử lý hữu ích là chạy dữ liệu thông qua một số quy trình làm sạch dữ liệu.

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.2. Tích hợp dữ liệu (Data intergration)

- Là tích hợp nhiều nguồn cơ sở dữ liệu (databases), khối dữ liệu (data cubes) hoặc tập tin (files) về chung 1 CSDL.
- Một số vấn đề gặp phải:
  - **Cùng một thuộc tính** đại diện cho một khái niệm nhất định **có thể có tên khác nhau** trong các cơ sở dữ liệu khác nhau. Ví dụ: thuộc tính custom_id trong CSDL này và id-cust trong CSDL khác.
  - **Sự không nhất quán** (inconsistencies) khi thu thập dữ liệu cũng có thể xảy ra. Ví dụ: cùng 1 khách hàng, cùng 1 số điện thoại nhưng tên có thể được đăng ký là "Bill" trong CSDL 1, "William" trong CSDL 2 và "Tom" trong CSDL 3.
  - **Dư thừa** (redundancies) Một số thuộc tính có thể được suy ra từ những thuộc tính khác (ví dụ: doanh thu hàng năm). Thuộc tính này có trong CSDL này và không có trong CSDL khác.

  $\Rightarrow$ Thông thường: việc làm sạch dữ liệu và tích hợp dữ liệu được thực hiện như một bước tiền xử lý khi chuẩn bị dữ liệu cho kho dữ liệu.

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.3. Giảm dữ liệu (Data reduction)

- Sau khi làm sạch và tích hợp dữ liệu, lúc này tập dữ liệu đã chọn để phân tích trở nên LỚN (*huge*) ⇒ chắc chắn sẽ làm chậm quá trình khai thác.
- Giảm thiểu dữ liệu tốt là giảm được kích thước tập dữ liệu nhưng vẫn tạo ra kết quả phân tích giống nhau (hoặc gần như giống nhau).
- Các chiến lược giảm thiểu dữ liệu bao gồm:
  - ***Giảm kích thước*** (*Dimensionality reduction*): các sơ đồ mã hóa dữ liệu được áp dụng để giảm bớt hoặc "nén" của dữ liệu gốc. Gồm các kỹ thuật nén dữ liệu như:
    - Sử dụng biến đổi wavelet (*wavelet transforms*)
    - Phân tích thành phần chính (*principal components analysis*)
    - Lựa chọn tập hợp con thuộc tính (*attribute subset selection* - như loại bỏ các thuộc tính không liên quan)
    - Xây dựng thuộc tính (*attribute construction* - như một tập hợp nhỏ các thuộc tính hữu ích hơn được lấy từ tập hợp ban đầu).

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.3. Giảm dữ liệu (Data reduction)

- Các chiến lược giảm thiểu dữ liệu bao gồm: (tt)
  - ***Giảm số lượng*** (*Numerosity reduction)*: dữ liệu được thay thế bằng các biểu diễn thay thế, nhỏ hơn bằng cách sử dụng các mô hình tham số, như:
    - Mô hình hồi quy (*regression*)
    - Mô hình log-tuyến tính (*log-linear models*)
    - Mô hình phi tham số (*nonparametric models* - như: biểu đồ (*histograms*), cụm (*clusters*), lấy mẫu (*sampling*)
    - Tổng hợp dữ liệu (*data aggregation*).

1.2.- Nhiệm vụ chính trong tiền xử lý dữ liệu

1.2.4. Chuyển đổi dữ liệu (Data transformation)

Các hình thức chuyển đổi dữ liệu:

- Chuẩn hóa (*normalization*)
- Rời rạc hóa dữ liệu (*data discretization*)
- Tạo phân cấp khái niệm (*concept hierarchy generation*)

NỘI DUNG CHƯƠNG 3

1. Tổng quan về xử lý dữ liệu (*Data Preprocessing: An Overview*)
2. Làm sạch dữ liệu (*Data Cleaning*)
3. Tích hợp dữ liệu (*Data Integration*)
4. Giảm thiểu dữ liệu (*Data Reduction*)
5. Chuyển đổi dữ liệu và phân tách dữ liệu (*Data Transformation and Data Discretization*)
6. Thực hành tiền xử lý dữ liệu

2. LÀM SẠCH DỮ LIỆU *(Data Cleaning)*

- Các quy trình làm sạch dữ liệu (*Data cleaning* - hoặc dọn dẹp dữ liệu - *Data cleansing*) cố gắng:
  - Điền vào các giá trị còn thiếu
  - Làm giảm nhiễu trong khi xác định các giá trị ngoại lệ
  - Sửa các điểm không nhất quán trong dữ liệu.
- Quá trình làm sạch dữ liệu (Data Cleaning as a Process)

  Quy trình 2 bước để làm sạch dữ liệu:
  - ***B1***: Phát hiện sự khác biệt.
  - ***B2***: Chuyển đổi dữ liệu để sửa chữa sự khác biệt của dữ liệu.

Bước 1: Phát hiện sự khác biệt

- Sự khác biệt có thể do một số yếu tố gây ra như:
  - Các biểu mẫu nhập dữ liệu được thiết kế kém có nhiều trường tùy chọn.
  - Lỗi của con người khi nhập dữ liệu:
    - Lỗi cố ý (VD: người trả lời không muốn tiết lộ thông tin về bản thân)
    - Dữ liệu lỗi thời (ví dụ: địa chỉ đã thay đổi nhưng không được cập nhật).
  - Lỗi do việc trình bày dữ liệu không nhất quán và việc sử dụng mã không nhất quán (ví dụ: “2010/12/25” và “25/12/2010” cho kiểu dữ liệu ngày).
  - Lỗi trong thiết bị đo đạc ghi lại dữ liệu.
  - Dữ liệu được sử dụng (không đầy đủ) cho các mục đích khác với dự định ban đầu.
  - Có thể có sự mâu thuẫn do tích hợp dữ liệu (ví dụ: cùng một thuộc tính nhưng có thể có các tên khác nhau trong các cơ sở dữ liệu khác nhau).

Bước 1: Phát hiện sự khác biệt

- Tiến hành phát hiện sự khác biệt:
  - Sử dụng bất kỳ kiến thức nào đã có về các thuộc tính của dữ liệu. Ví dụ:
    - Kiểu dữ liệu và miền giá trị của từng thuộc tính là gì?
    - Các giá trị được chấp nhận cho mỗi thuộc tính là gì?
    - Xác định mean, median, mode để nắm bắt xu hướng dữ liệu và xác định các điểm bất thường.
    - Dữ liệu có đối xứng hay bị lệch không?
    - Phạm vi của các giá trị là gì?
    - Tất cả các giá trị có nằm trong phạm vi dự kiến không? Độ lệch chuẩn của từng thuộc tính là bao nhiêu?
    - Có bất kỳ sự phụ thuộc nào đã biết giữa các thuộc tính không?
- Có thể sử dụng chương trình tự viết và/hoặc sử dụng một số công cụ để có thể tìm thấy nhiễu, giá trị ngoại lệ và giá trị bất thường.

Bước 1: Phát hiện sự khác biệt

- Kiểm tra dữ liệu về các quy tắc:
  - ***Quy tắc Duy nhất*** (*unique rules*): cho biết mỗi giá trị của thuộc tính đã cho phải khác với tất cả các giá trị khác của thuộc tính đó
  - ***Quy tắc Liên tiếp*** (*consecutive rules*): không được thiếu giá trị nào giữa giá trị thấp nhất và cao nhất của thuộc tính và tất cả các giá trị cũng phải là duy nhất
  - ***Quy tắc Vô hiệu*** (*unique rules*, *consecutive rules*, and *null rules*): chỉ định việc sử dụng khoảng trống, dấu chấm hỏi, ký tự đặc biệt hoặc các chuỗi khác có thể chỉ ra điều kiện null (ví dụ: khi một thuộc tính không có giá trị sẽ được gán =0 hay "?" hoặc "không biết", …) và cách xử lý các giá trị đó.

Bước 2: Chuyển đổi dữ liệu để sửa chữa sự khác biệt của dữ liệu

- Sử dụng công cụ: Hiện có nhiều công cụ hỗ trợ việc phát hiện và sửa chữa sự khác biệt:
  - Đơn giản như:
    - Việc "*Find and Replace*" (VD: thay thế chuỗi "gender" bằng "sex")
    - Các công cụ lọc dữ liệu sử dụng kiến thức miền đơn giản (ví dụ: kiến thức về địa chỉ bưu chính và kiểm tra chính tả) để phát hiện lỗi và sửa dữ liệu.
  - Các công cụ ETL (*extraction/transformation/loading* - trích xuất/chuyển đổi/tải) cho phép người dùng chỉ định các phép biến đổi thông qua giao diện người dùng đồ họa (GUI).
  - Các công cụ này dựa vào kỹ thuật phân tích cú pháp (*parsing*) và kết hợp mờ (*fuzzy matching*).
  - Các công phân tích dữ liệu để khám phá các quy tắc và mối quan hệ, đồng thời phát hiện dữ liệu vi phạm các điều kiện đó.

NỘI DUNG CHƯƠNG 3

1. Tổng quan về xử lý dữ liệu (*Data Preprocessing: An Overview*)
2. Làm sạch dữ liệu (*Data Cleaning*)
3. Tích hợp dữ liệu (*Data Integration*)
4. Giảm thiểu dữ liệu (*Data Reduction*)
5. Chuyển đổi dữ liệu và phân tách dữ liệu (*Data Transformation and Data Discretization*)
6. Thực hành tiền xử lý dữ liệu

3. TÍCH HỢP DỮ LIỆU *(Data Integration)*

Khai thác dữ liệu thường yêu cầu hợp nhất dữ liệu từ nhiều kho dữ liệu. Việc tích hợp cẩn thận có thể giúp giảm thiểu và tránh sự dư thừa cũng như sự không nhất quán trong tập dữ liệu thu được. Điều này có thể giúp cải thiện độ chính xác và tốc độ của quá trình khai thác dữ liệu tiếp theo.

3.1. Nhận dạng thực thể *(Entity Identification Problem)*

- Là làm cách nào để có thể khớp các thực thể trong thế giới thực tương đương từ nhiều nguồn dữ liệu? Ví dụ: làm thế nào nhà phân tích dữ liệu hoặc máy tính có thể chắc chắn rằng
- id_Customer khách hàng trong một CSDL và Manager_Id trong CSDL khác tham chiếu đến cùng một thuộc tính?
- Mã dữ liệu cho loại thanh toán trong một cơ sở dữ liệu có thể là “H” và “S” nhưng lại có giá trị là 1 và 2 trong cơ sở dữ liệu khác
- Phạm vi miền giá trị của thuộc tính đó?
- Quy tắc null đang được áp dụng để xử lý các giá trị trống, 0 hoặc null?
- Các ràng buộc tham chiếu trong hệ thống nguồn đều khớp với các thuộc tính trong hệ thống đích. VD: trong một hệ thống, chiết khấu có thể được áp dụng cho đơn đặt hàng, trong khi ở hệ thống khác, chiết khấu được áp dụng cho từng chi tiết đơn hàng riêng lẻ trong đơn đặt hàng.
- …

  ⇒ Bước này cũng liên quan đến việc làm sạch dữ liệu, như được mô tả trước đó.

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

- Dư thừa (*Redundancy*) là một vấn đề quan trọng khác trong tích hợp dữ liệu. Một thuộc tính (chẳng hạn như doanh thu hàng năm) có thể dư thừa nếu nó có thể được tính toán (*derived*) từ một hoặc tập hợp các thuộc tính khác.
- Sự không nhất quán trong cách đặt tên thuộc tính cũng có thể gây ra sự dư thừa trong tập dữ liệu kết quả.
- Một số dư thừa có thể được phát hiện bằng phân tích tương quan (*correlation analysis*). Với hai thuộc tính, phân tích như vậy có thể đo lường mức độ ảnh hưởng của một thuộc tính đến thuộc tính kia dựa trên dữ liệu có sẵn.

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.1. Phép kiểm tra tương quan χ^2 cho dữ liệu danh nghĩa

(χ^2 Correlation Test for Nominal Data)

– Đối với dữ liệu danh nghĩa, mối quan hệ tương quan giữa hai thuộc tính A và B có thể được phát hiện bằng phép kiểm tra χ^2 (*chi-square*). Giả sử A có c giá trị phân biệt, cụ thể là $a_1, a_2, ..., a_c$. B có r giá trị riêng biệt, cụ thể là $b_1, b_2, ..., b_r$. Các bộ dữ liệu được mô tả bởi A và B có thể được hiển thị dưới dạng bảng thống kê, với các giá trị c của A tạo thành các cột và giá trị r của B tạo thành các hàng. Gọi (A_i, B_j) biểu thị sự kiện chung mà thuộc tính A nhận giá trị a_i và thuộc tính B nhận giá trị b_j, nghĩa là trong đó $(A = a_i, B = b_j)$. Mọi cặp giá trị chung có thể có (A_i, B_j) đều có ô (hoặc vị trí) riêng trong bảng.

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.1. Phép kiểm tra tương quan χ^2 cho dữ liệu danh nghĩa (χ^2 Correlation Test for Nominal Data)

– Giá trị χ^2 (còn được gọi là thống kê *Pearson* χ^2) được tính như sau:

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij}-e_{ij})^2}{e_{ij}} \qquad \textit{Công thức 1}$$

trong đó

- o_{ij} là tần suất quan sát được (tức là số lượng thực tế) của sự kiện chung (A_i, B_j).
- e_{ij} là tần suất dự kiến của (A_i, B_j), có thể được tính như sau

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij}-e_{ij})^2}{e_{ij}} \qquad \textit{Công thức 2}$$

với: + n là số bộ dữ liệu;
+ $count(A = a_i)$ là số bộ có giá trị a_i cho A;
+ $count(B = b_j)$ là số bộ có giá trị b_j cho B.

– Tổng trong công thức (1) được tính trên tất cả các ô $r \times c$.

– Lưu ý rằng các ô đóng góp nhiều nhất vào giá trị χ^2 là những ô có số lượng thực tế rất khác so với dự kiến.

– Thống kê χ^2 kiểm định giả thuyết A và B độc lập, nghĩa là giữa chúng không có mối tương quan. Việc kiểm tra dựa trên mức ý nghĩa, với $(r-1)\times(c-1)$ bậc tự do.

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.1. Phép kiểm tra tương quan χ^2 cho dữ liệu danh nghĩa

– Ví dụ : Giả sử có khảo sát 1500 người để thăm dò xem loại tài liệu đọc ưa thích của họ là hư cấu hay phi hư cấu. Tần suất (số lượng) quan sát được của mỗi sự kiện chung có thể xảy ra được tóm tắt trong bảng.

	male	*female*	*Total*
fiction	250	200	450
non_fiction	50	1000	1050
Total	300	1200	1500

Sử dụng công thức (2), ta có thể tính được tần suất dự kiến cho từng ô (các số trong ngoặc đơn). Ví dụ: tần suất dự kiến cho ô (*male*, *fiction*) là:

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

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

Sử dụng công thức (1) để tính χ^2, ta có

$$\chi^2 = \sum_{i=1}^{c}\sum_{j=1}^{r}\frac{(o_{ij}-e_{ij})^2}{e_{ij}} = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840}$$

$$= 284.44 + 121.90 + 71.11 + 30.48 = 507.93$$

Đối với bảng 2×2 này, bậc tự do (*degrees of freedom*) là (2−1)(2−1) = 1. Đối với 1 bậc tự do, giá trị χ^2 cần thiết để bác bỏ giả thuyết ở mức ý nghĩa 0,001 là 10.828 (lấy từ bảng điểm phần trăm trên của phân bố χ^2, thường có trong bất kỳ sách giáo khoa nào về thống kê). Vì giá trị tính toán cao hơn giá trị này nên có thể bác bỏ giả thuyết rằng giới tính và cách đọc ưa thích là độc lập và kết luận rằng hai thuộc tính này có mối tương quan (mạnh mẽ) đối với một nhóm người nhất định.

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.2. Hệ số tương quan cho dữ liệu kiểu số (Correlation Coefficient for Numeric Data)

– Có thể đánh giá mối tương quan giữa hai thuộc tính kiểu số A và B bằng cách tính hệ số tương quan (còn được gọi là *hệ số mômen sản phẩm Pearson - Pearson's product moment coefficient* -, được đặt theo tên của người phát minh ra nó, *Karl Pearson*) theo công thức sau:

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

Công thức 2-3

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.2. Hệ số tương quan cho dữ liệu kiểu số (Correlation Coefficient for Numeric Data)

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

trong đó

- n là số bộ dữ liệu.
- a_i và b_i là giá trị tương ứng của A và B trong bộ i,
- $\bar{A}$ và $\bar{B}$ là giá trị trung bình tương ứng của A và B.
- σ_A và σ_B là độ lệch chuẩn (*standard deviations*) tương ứng của A và B.
- $\Sigma(a_i b_i)$ là tổng của các tích giữa A và B (tức là, đối với mỗi bộ dữ liệu, giá trị của A được nhân với giá trị của B trong bộ dữ liệu đó).

Lưu ý rằng $-1 \leq r_{A,B} \leq +1$. Nếu:

- $r_{A,B} > 0$: A và B có mối tương quan dương, nghĩa là giá trị của A tăng khi giá trị của B tăng. Giá trị càng cao thì mối tương quan càng mạnh (nghĩa là mỗi thuộc tính càng hàm ý thuộc tính kia càng nhiều). Do đó, giá trị cao hơn có thể chỉ ra rằng A (hoặc B) có thể bị loại bỏ do dư thừa.
- $r_{A,B} = 0$: A và B độc lập và không có mối tương quan giữa chúng.
- $r_{A,B} < 0$: A và B có mối tương quan nghịch, trong đó giá trị của một thuộc tính tăng khi giá trị của thuộc tính kia giảm.

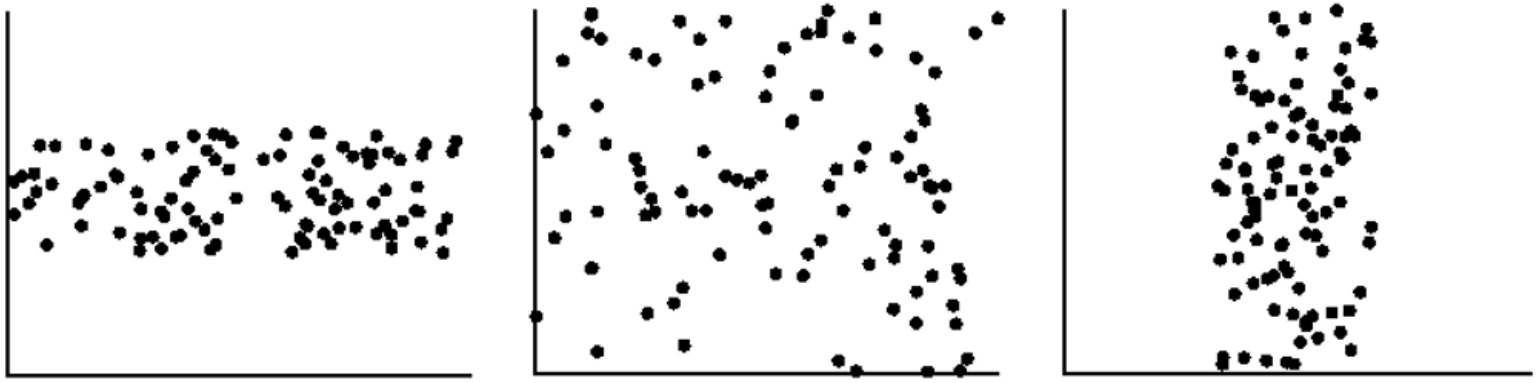
3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.2. Hệ số tương quan cho dữ liệu kiểu số (Correlation Coefficient for Numeric Data)

– Minh họa sử dụng biểu đồ phân tán (*Scatter plots*) để xem mối tương quan giữa các thuộc tính.

(a) Dữ liệu tương quan dương giữa A và B

(b) Dữ liệu tương quan âm giữa A và B



(c) Dữ liệu A và B không tương quan

Lưu ý: mối tương quan không hàm ý quan hệ nhân quả. Nghĩa là, nếu A và B tương quan với nhau thì điều này không nhất thiết ngụ ý rằng A gây ra B hoặc B gây ra A.

Ví dụ: khi phân tích cơ sở dữ liệu nhân khẩu học, có thể thấy các thuộc tính số bệnh viện và số vụ trộm xe trong một khu vực có mối tương quan với nhau. Điều này không có nghĩa là cái này gây ra cái kia. Cả hai đều thực sự có mối liên hệ nhân quả với thuộc tính thứ ba, cụ thể là dân số.

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.3. Hiệp phương sai của dữ liệu số (Covariance of Numeric Data)

Trong lý thuyết xác suất và thống kê, mối tương quan (*correlation*) và hiệp phương sai (*covariance*) là hai thước đo tương tự nhau để đánh giá mức độ thay đổi của hai thuộc tính cùng nhau. Xét hai thuộc tính số ***A*** và ***B*** và một tập hợp ***n*** quan sát $\{(a_1, b_1), \ldots, (a_n, b_n)\}$. Giá trị trung bình của ***A*** và ***B*** tương ứng còn được gọi là giá trị kỳ vọng (*expected values*) của ***A*** và ***B***, nghĩa là

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} \qquad \text{và} \qquad E(B) = \bar{B} = \frac{\sum_{i=1}^{n} b_i}{n}$$

- Hiệp phương sai giữa A và B được định nghĩa theo công thức sau:

$$Cov(A, B) = E((A - \bar{A})(B - \bar{B})) = \frac{\sum_{i=1}^{n}(a_i - \bar{A})-(b_i - \bar{B})}{n} \qquad \textit{Công thức 4}$$

- So sánh công thức (3) đối với $r_{A,B}$ (hệ số tương quan), với công thức (4) đối với hiệp phương sai, ta thấy

$$r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B} \qquad \textit{Công thức 5}$$

- trong đó σA và σB lần lượt là độ lệch chuẩn của A và B. Nó cũng có thể được chỉ ra rằng

$$Cov(A, B) = E(A \times B) - \bar{A}\bar{B} \qquad \textit{Công thức 6}$$

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.3. Hiệp phương sai của dữ liệu số (Covariance of Numeric Data)

– Đối với hai thuộc tính A và B có xu hướng thay đổi cùng nhau, nếu A lớn hơn $\bar{A}$ (giá trị kỳ vọng của A) thì B có khả năng lớn hơn $\bar{B}$ (giá trị kỳ vọng của B). Do đó, hiệp phương sai giữa A và B là dương. Mặt khác, nếu một trong các thuộc tính có xu hướng cao hơn giá trị mong đợi của nó trong khi thuộc tính kia thấp hơn giá trị mong đợi của nó thì hiệp phương sai của A và B là âm.

Nếu A và B độc lập (*independent* - tức là chúng không có tương quan) thì

$$E(A \times B) = E(A) \times E(B)$$

Do đó, hiệp phương sai là $Cov(A, B) = E(A \times B) - \bar{A} \times \bar{B} = E(A) \times E(B) - \bar{A} \times \bar{B} = 0$

Tuy nhiên, điều ngược lại là không đúng. Một số cặp biến ngẫu nhiên (thuộc tính) có thể có hiệp phương sai bằng 0 nhưng không độc lập. Chỉ theo một số giả định bổ sung

3.2. Phân tích dư thừa và tương quan *(Redundancy and Correlation Analysis)*

3.2.3. Hiệp phương sai của dữ liệu số (Covariance of Numeric Data)

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

– Ví dụ: Cho bảng trình bày một ví dụ đơn giản về giá cổ phiếu được quan sát tại năm thời điểm của 2 công ty A và X. Nếu các cổ phiếu bị ảnh hưởng bởi cùng xu hướng của ngành, giá của chúng sẽ tăng hay giảm cùng nhau?

$$E(A) = \bar{A} = \frac{\sum_{i=1}^{n} a_i}{n} = \frac{6+5+4+3+2}{5} = \frac{20}{5} = 4$$

và $$E(X) = \bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} = \frac{20+10+14+5+5}{5} = \frac{54}{5} = 10.8$$

Sử dụng công thức 6, ta có:

$$\text{Cov(A, X)} = E(A \times X) - (\bar{\text{A}} \times \bar{\text{X}}) = \frac{(6 \times 20) + (5 \times 10) + (4 \times 14) + (3 \times 5) + (2 \times 5)}{5} - (4 \times 10.8) = 50.2 - 43.2 = 7$$

$\Rightarrow$ Với hiệp phương sai dương, có thể nói rằng giá cổ phiếu của cả hai công ty đều tăng cùng nhau.

Phương sai là trường hợp đặc biệt của hiệp phương sai, trong đó hai thuộc tính giống hệt nhau (nghĩa là hiệp phương sai của một thuộc tính với chính nó).

3.3. Trùng lặp dữ liệu *(Tuple Duplication)*

- Ngoài việc phát hiện sự dư thừa giữa các thuộc tính, sự trùng lặp cũng cần được phát hiện ở cấp độ bộ dữ liệu (tức là trùng lặp giữa các dòng trong dữ liệu).
- Sự không nhất quán thường phát sinh giữa các bản sao khác nhau, do nhập dữ liệu không chính xác hoặc cập nhật một số lần nhưng không phải cập nhật tất cả dữ liệu.

Ví dụ: số điện thoại của cùng một người mua xuất hiện với các địa chỉ khác nhau trong cơ sở dữ liệu đơn đặt hàng.

3.4. Phát hiện và giải quyết xung đột giá trị dữ liệu *(Data Value Conflict Detection and Resolution)*

– Tích hợp dữ liệu cũng liên quan đến việc phát hiện và giải quyết xung đột giá trị dữ liệu.

Ví dụ: đối với cùng một thực thể trong thế giới thực, các giá trị thuộc tính từ các nguồn khác nhau có thể khác nhau. Điều này có thể là do sự khác biệt trong cách trình bày, chia tỷ lệ hoặc mã hóa.

– Các thuộc tính cũng có thể khác nhau về mức độ trừu tượng, trong đó một thuộc tính trong một hệ thống được ghi lại ở mức độ trừu tượng thấp hơn thuộc tính "tương tự" trong một hệ thống khác.

Ví dụ: trường A xếp loại học sinh theo các mức: ***xuất sắc***, giỏi, khá, trung bình, kém, ***yếu***; nhưng trường Y lại xếp loại học sinh theo các mức: giỏi, khá, ***trung bình-khá***, trung bình, kém.

3.4. Phát hiện và giải quyết xung đột giá trị dữ liệu *(Data Value Conflict Detection and Resolution)*

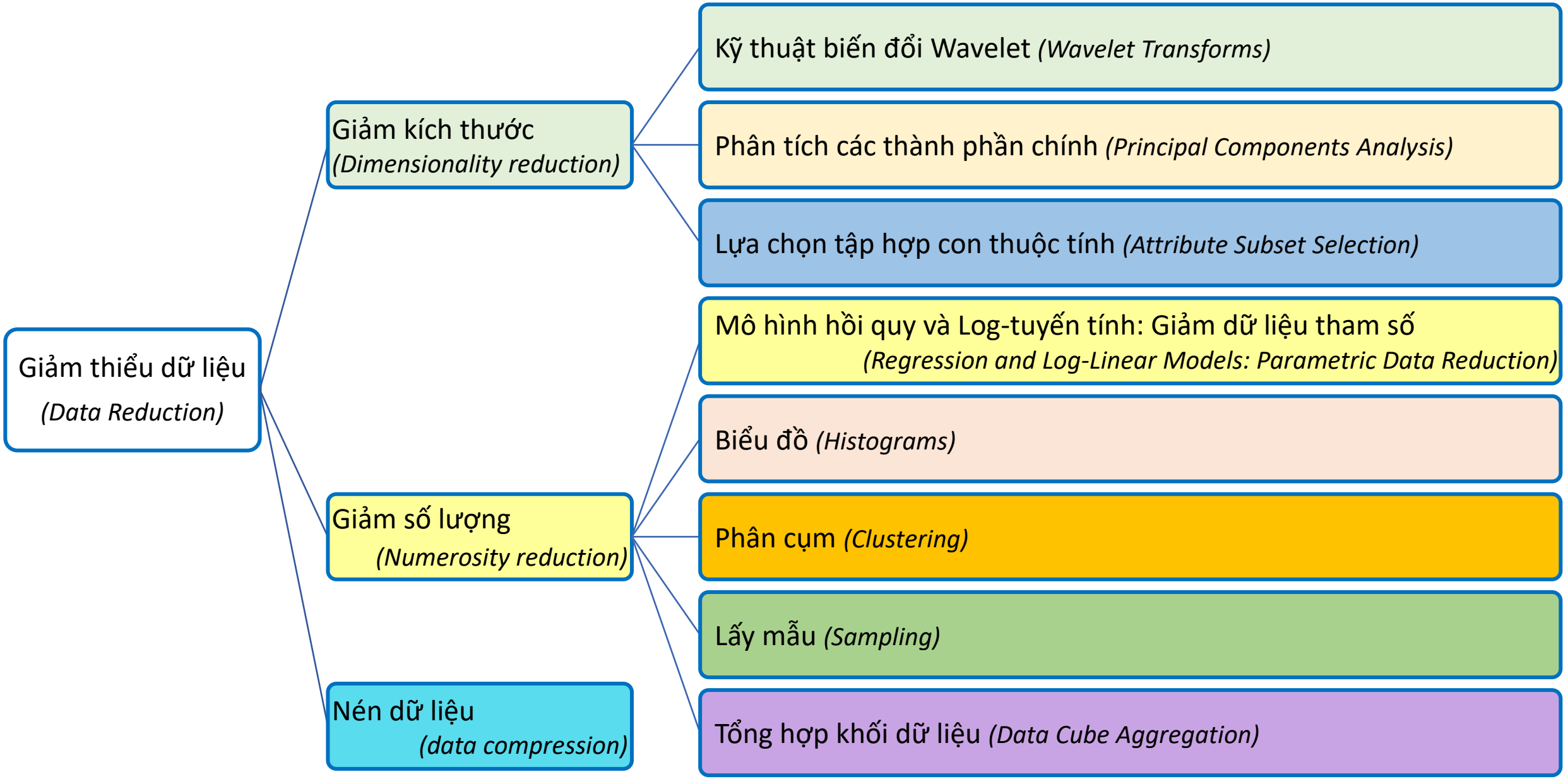
– Một số ví dụ:

- Thuộc tính đơn vị tính trong hệ thống X sử dụng là Kg, nhưng trong hệ thống Y sử dụng đơn vị tính là pounds.
- Đối với một chuỗi khách sạn, giá phòng ở các thành phố khác nhau có thể không chỉ liên quan đến các loại tiền tệ khác nhau mà còn liên quan đến các dịch vụ khác nhau (ví dụ: bữa sáng miễn phí) và thuế.
- Khi trao đổi thông tin giữa các trường, mỗi trường có thể có chương trình giảng dạy và hệ thống chấm điểm riêng. Một trường đại học X có thể áp dụng hệ thống 4 học kỳ/ năm học, và chấm điểm theo thang điểm từ A+ đến F, trong khi trường đại học Y có thể áp dụng hệ thống 2 học kỳ/năm học và chấm điểm theo thang điểm từ 1 đến 10. Rất khó để làm việc đưa ra các quy tắc chuyển đổi chính xác giữa hai trường đại học, khiến việc trao đổi thông tin trở nên khó khăn.

NỘI DUNG CHƯƠNG 3

1. Tổng quan về xử lý dữ liệu (*Data Preprocessing: An Overview*)
2. Làm sạch dữ liệu (*Data Cleaning*)
3. Tích hợp dữ liệu (*Data Integration*)
4. Giảm thiểu dữ liệu (*Data Reduction*)
5. Chuyển đổi dữ liệu và phân tách dữ liệu (*Data Transformation and Data Discretization*)
6. Thực hành tiền xử lý dữ liệu

4. GIẢM THIỂU DỮ LIỆU *(Data Reduction)*



4.1. Kỹ thuật biến đổi Wavelet (Wavelet Transforms)

- Các phép biến đổi *Wavelet* có thể được áp dụng cho dữ liệu đa chiều như khối dữ liệu (*data cube*).
- Điều này được thực hiện bằng cách trước tiên áp dụng phép biến đổi cho chiều thứ nhất, sau đó đến chiều thứ hai, v.v…
- Độ phức tạp tính toán liên quan là tuyến tính đối với số lượng ô trong khối.
- Các phép biến đổi *Wavelet* cho kết quả tốt trên dữ liệu thưa thớt hoặc bị lệch và trên dữ liệu có thuộc tính được sắp xếp.
- Nén tổn hao (*lossy compression*) bằng *Wavelet* được cho là tốt hơn nén JPEG (tiêu chuẩn thương mại hiện tại).
- Biến đổi *Wavelet* có nhiều ứng dụng trong thế giới thực, bao gồm nén hình ảnh dấu vân tay, thị giác máy tính, phân tích dữ liệu chuỗi thời gian và làm sạch dữ liệu.

4.1. Kỹ thuật biến đổi Wavelet (Wavelet Transforms)

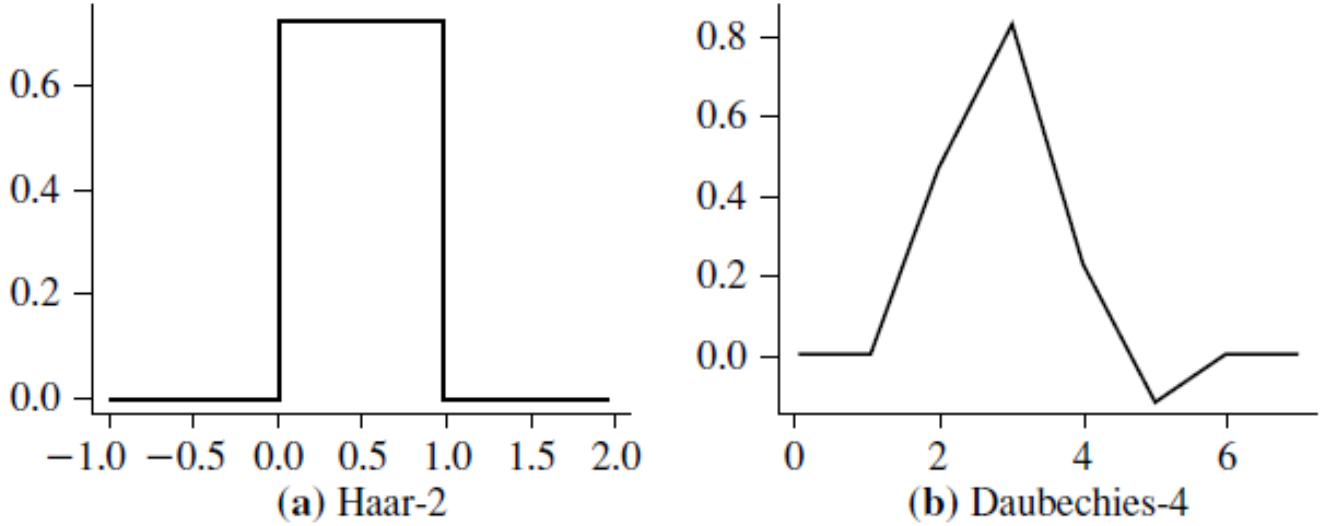
- Biến đổi sóng con rời rạc (*discrete wavelet transform - DWT*) là một kỹ thuật xử lý tín hiệu tuyến tính, khi áp dụng cho vectơ dữ liệu X, sẽ biến đổi nó thành một vectơ khác (X_0) về mặt số lượng của các hệ số sóng con (*wavelet coefficients*).
- Hai vectơ có cùng độ dài. Khi áp dụng kỹ thuật này để giảm dữ liệu, mỗi bộ dữ liệu sẽ được xem là một vectơ dữ liệu ***n*** chiều, nghĩa là $X = (x_1, x_2, ..., x_n)$, mô tả n phép đo được thực hiện trên bộ dữ liệu từ n thuộc tính.
- Dữ liệu gần đúng đã được nén có thể được giữ lại bằng cách chỉ lưu trữ một phần nhỏ hệ số sóng con mạnh nhất.

  Ví dụ: tất cả các hệ số *wavelet* lớn hơn một số ngưỡng do người dùng chỉ định có thể được giữ lại. Tất cả các hệ số khác được đặt thành *0*.

  $\Rightarrow$ việc biểu diễn dữ liệu thu được rất thưa thớt, nhờ vậy các thao tác có thể tận dụng tính thưa thớt của dữ liệu sẽ được tính toán rất nhanh nếu được thực hiện trong không gian *wavelet*.

4.1. Kỹ thuật biến đổi Wavelet (Wavelet Transforms)

- Kỹ thuật này cũng có tác dụng loại bỏ nhiễu mà không làm mịn các tính năng chính của dữ liệu, giúp việc làm sạch dữ liệu trở nên hiệu quả.
- Các phép biến đổi *wavelet* phổ biến bao gồm *Haar-2*, *Daubechies-4* và *Daubechies-6*.



(a) Haar-2 (b) Daubechies-4

- Phương pháp thực hiện:
  i. Độ dài L của vectơ dữ liệu đầu vào phải có giá trị là lũy thừa nguyên của 2. Điều kiện này có thể được đáp ứng bằng cách đệm vectơ dữ liệu bằng các số 0 nếu cần ($L \geq n$).

4.1. Kỹ thuật biến đổi Wavelet (Wavelet Transforms)

- Phương pháp này như sau:

ii. Mỗi phép biến đổi liên quan đến việc áp dụng hai hàm.
   - Hàm đầu tiên áp dụng một số thao tác làm mịn dữ liệu, chẳng hạn như tính tổng hoặc trung bình có trọng số.
   - Hàm thứ hai thực hiện sự khác biệt có trọng số, có tác dụng làm nổi bật các tính năng chi tiết của dữ liệu.

iii. Hai hàm này được áp dụng cho các cặp điểm dữ liệu trong X, nghĩa là cho tất cả các cặp số đo (x_{2i},x_{2i+1}). Điều này dẫn đến hai bộ dữ liệu có độ dài L/2. Nói chung, chúng thể hiện phiên bản được làm mịn (*smoothed*) hoặc tần suất thấp (*low-frequency*) của dữ liệu đầu vào và nội dung tần số cao (*high-frequency*) của nó tương ứng.

iv. Hai hàm này được áp dụng đệ quy cho các tập dữ liệu thu được ở vòng lặp trước, cho đến khi các tập dữ liệu thu được có độ dài 2.

v. Các giá trị được chọn từ các tập dữ liệu thu được trong các lần lặp trước được chỉ định là hệ số *wavelet* của dữ liệu được chuyển đổi.

4.2. Phân tích các thành phần chính (Principal Components Analysis -PCA)

- Phân tích các thành phần chính còn được gọi là phương pháp *Karhunen-Loeve*, hoặc *K-L*
- Phân tích các thành phần chính như một phương pháp giảm kích thước của dữ liệu.
- *PCA* có thể được áp dụng cho:
  - Các thuộc tính có thứ tự và không có thứ tự
  - Dữ liệu thưa thớt (*sparse data*)
  - Dữ liệu sai lệch (*skewed data*)
  - *Dữ liệu đa chiều* (*Multidimensional data* - có nhiều hơn hai chiều) có thể được xử lý bằng cách giảm số chiều xuống còn hai chiều.
- Các thành phần chính có thể được sử dụng làm đầu vào cho phân tích cụm và hồi quy bội.
- So với các phép biến đổi *wavelet*, *PCA* có xu hướng xử lý dữ liệu thưa thớt tốt hơn, trong khi các phép biến đổi *wavelet* phù hợp hơn với dữ liệu có nhiều chiều.

4.2. Phân tích các thành phần chính *(Principal Components Analysis -PCA)*

- Giả sử rằng dữ liệu được giảm bao gồm các bộ dữ liệu hoặc vectơ dữ liệu được mô tả bởi n thuộc tính (hoặc kích thước n). Phân tích thành phần chính tìm kiếm vectơ trực giao *k* cho *n* chiều (*n-dimensional*) có thể được sử dụng tốt nhất để biểu diễn dữ liệu, trong đó $k \leq n$. Do đó, dữ liệu gốc được chiếu lên một không gian nhỏ hơn nhiều, dẫn đến giảm kích thước. *PCA* "kết hợp" bản chất của các thuộc tính bằng cách tạo ra một tập hợp biến thay thế nhỏ hơn. Dữ liệu ban đầu sau đó có thể được chiếu lên tập nhỏ hơn này. *PCA* thường tiết lộ được những mối quan hệ mà trước đây không bị nghi ngờ và do đó cho phép diễn giải những điều mà thông thường sẽ không dẫn đến kết quả.

4.2. Phân tích các thành phần chính *(Principal Components Analysis -PCA)*

- Quy trình cơ bản của Phân tích các thành phần chính:

i. Dữ liệu đầu vào được chuẩn hóa để mỗi thuộc tính nằm trong cùng một phạm vi. Bước này giúp đảm bảo rằng các thuộc tính có miền giá trị lớn sẽ không lấn át các thuộc tính có miền giá trị nhỏ hơn.

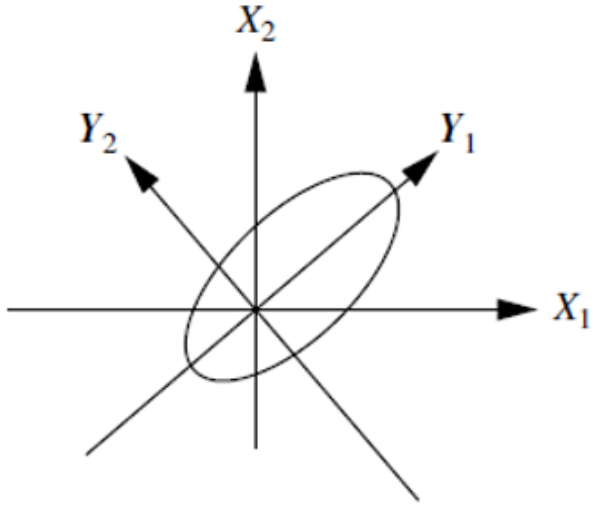
ii. *PCA* tính toán *k* vectơ trực giao (*orthonormal vectors*) làm cơ sở cho dữ liệu đầu vào được chuẩn hóa. Đây là các vectơ đơn vị mà mỗi vectơ hướng vuông góc với các vectơ khác. Các vectơ này được gọi là các thành phần chính (*principal components*). Dữ liệu đầu vào là sự kết hợp tuyến tính của các thành phần chính.

iii. Các thành phần chính được sắp xếp giảm dần theo mức độ quan trọng của "*ý nghĩa*" hoặc "*độ mạnh*". Các thành phần chính về cơ bản đóng vai trò như một bộ trục mới cho dữ liệu, cung cấp thông tin quan trọng về phương sai. Nghĩa là, các trục được sắp xếp sao cho trục đầu tiên hiển thị phương sai lớn nhất trong số dữ liệu, trục thứ hai hiển thị phương sai cao nhất tiếp theo, v.v.

4.2. Phân tích các thành phần chính (Principal Components Analysis -PCA)

- Quy trình cơ bản của Phân tích các thành phần chính:

iv. Do các thành phần được sắp xếp theo thứ tự giảm dần về "mức độ quan trọng" (*significance*) nên kích thước dữ liệu có thể giảm bằng cách loại bỏ các thành phần yếu hơn (*weaker components*), tức là những thành phần có phương sai thấp. Bằng cách sử dụng các thành phần chính mạnh nhất, có thể xây dựng lại dữ liệu gốc gần đúng.



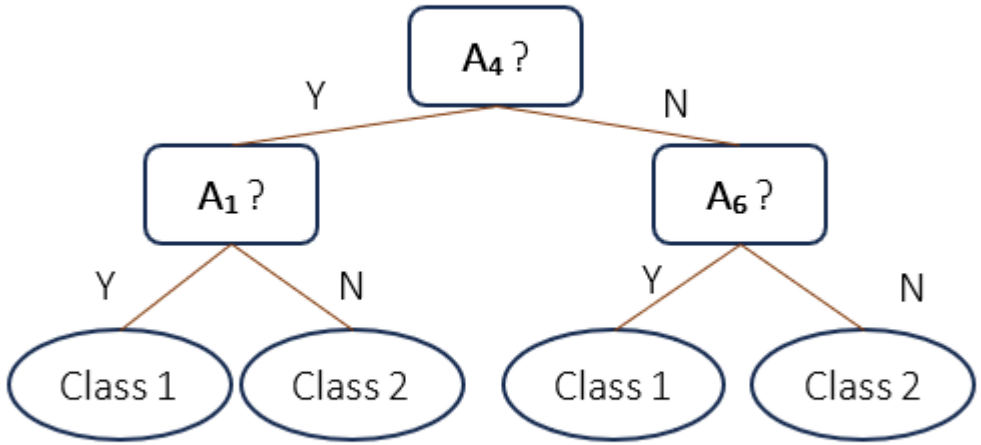
Hình 2-6.- Phân tích các thành phần chính.
Y1 và Y2 là hai thành phần chính đầu tiên của dữ liệu đã cho.

4.3. Lựa chọn tập hợp con thuộc tính (Attribute Subset Selection)

- Các bộ dữ liệu để phân tích có thể chứa hàng trăm thuộc tính, nhiều thuộc tính trong số đó có thể không liên quan đến nhiệm vụ khai thác hoặc dư thừa.
- Lựa chọn tập hợp con thuộc tính giúp giảm kích thước tập dữ liệu bằng cách loại bỏ các thuộc tính không liên quan hoặc dư thừa.
- ***Mục tiêu***:
  - Phân bố xác suất thu được của các thuộc tính được lựa chọn gần với phân bố khi sử dụng tất cả các thuộc tính.
  - Giảm số lượng thuộc tính xuất hiện trong các mẫu được phát hiện, giúp làm cho các mẫu này dễ hiểu hơn.
  - Việc giảm các thuộc tính không liên quan hoặc dư thừa còn giúp cải thiện thời gian khai thác

4.3. Lựa chọn tập hợp con thuộc tính (Attribute Subset Selection)

- Tập dữ liệu với n thuộc tính, có thể có 2^n tập con $\Rightarrow$ việc tìm kiếm toàn diện tập hợp con các thuộc tính tối ưu có thể cực kỳ tốn kém.
- Do đó, các phương pháp heuristic khám phá không gian tìm kiếm rút gọn thường được sử dụng để lựa chọn tập hợp con thuộc tính.
- Các phương pháp heuristic cơ bản của việc lựa chọn tập hợp con thuộc tính:

Forward selection	*Backward elimination*	*Decision tree induction*
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ ***Initial reduced set***: {} => $\{A_1\}$ => $\{A_1, A_4\}$ => **Reduced attribute set**: $\{A_1, A_4, A_6\}$	***Initial attribute set***: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_3, A_4, A_5, A_6\}$ => $\{A_1, A_4, A_5, A_6\}$ => ***Reduced attribute set***: $\{A_1, A_4, A_6\}$	***Initial attribute set***: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  => ***Reduced attribute set***: $\{A_1, A_4, A_6\}$

Các phương pháp tham lam (greedy - heuristic) để lựa chọn tập hợp con thuộc tính.

4.4. Mô hình hồi quy và Log-tuyến tính: Giảm dữ liệu tham số

(Regression and Log-Linear Models: Parametric Data Reduction)

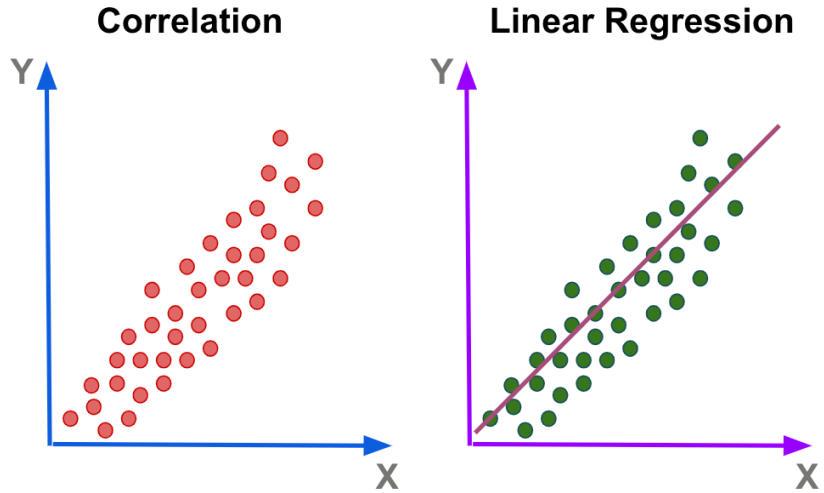
- Cả mô hình hồi quy và log-tuyến tính đều có thể được sử dụng trên dữ liệu thưa thớt, mặc dù ứng dụng của chúng có thể bị hạn chế.
- Mặc dù cả hai phương pháp đều có thể xử lý dữ liệu sai lệch (*skewed data*), nhưng hồi quy lại hoạt động rất tốt.
- Hồi quy có thể cần nhiều tính toán khi áp dụng cho dữ liệu nhiều chiều, trong khi các mô hình log-tuyến tính cho thấy khả năng mở rộng tốt cho tối đa 10 chiều.

4.4. Mô hình hồi quy và Log-tuyến tính: Giảm dữ liệu tham số (Regression and Log-Linear Models: Parametric Data Reduction)

4.4.1. Mô hình hồi quy (Regression model)

– Trong hồi quy tuyến tính (*linear regression* - đơn giản), dữ liệu được mô hình hóa để khớp với một phương trình đường thẳng. Ví dụ: một biến ngẫu nhiên, y (được gọi là *biến phản hồi* - *response variable*), có thể được mô hình hóa dưới dạng hàm tuyến tính của một biến ngẫu nhiên khác, x (được gọi là biến dự đoán - *predictor variable*), với phương trình:

$$y = wx + b$$ *Công thức 7*



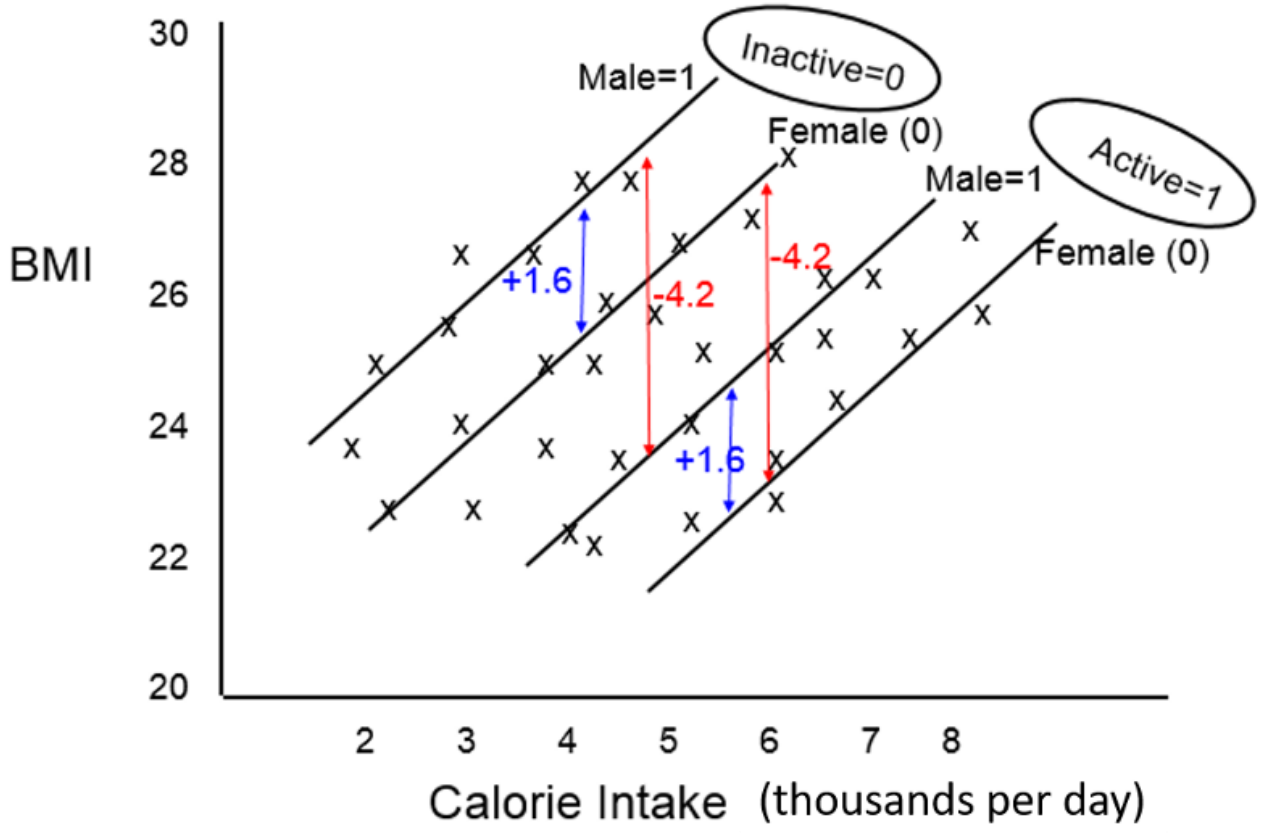
trong đó

- phương sai (*variance*) của y được coi là không đổi (*constant*).
- *x* và *y* là các thuộc tính kiểu số.
- Các hệ số *w* và *b* (được gọi là hệ số hồi quy - *regression coefficients*), chỉ định độ dốc của đường thẳng và điểm chặn *y* tương ứng. Các hệ số này có thể giải bằng phương pháp bình phương tối thiểu (*method of least squares*), giúp giảm thiểu sai số giữa đường thực tế phân tách dữ liệu (*the actual line separating the data*) và ước lượng của đường thẳng (*the estimate of the line*).

4.4. Mô hình hồi quy và Log-tuyến tính: Giảm dữ liệu tham số *(Regression and Log-Linear Models: Parametric Data Reduction)*

4.4.1. Mô hình hồi quy *(Regression model)*

- Hồi quy tuyến tính bội (*Multiple linear regression*) là phần mở rộng của hồi quy tuyến tính (đơn giản), cho phép một biến phản hồi, *y*, được mô hình hóa như một hàm tuyến tính (*linear function*) của hai hoặc nhiều biến dự đoán (*predictor variables*).

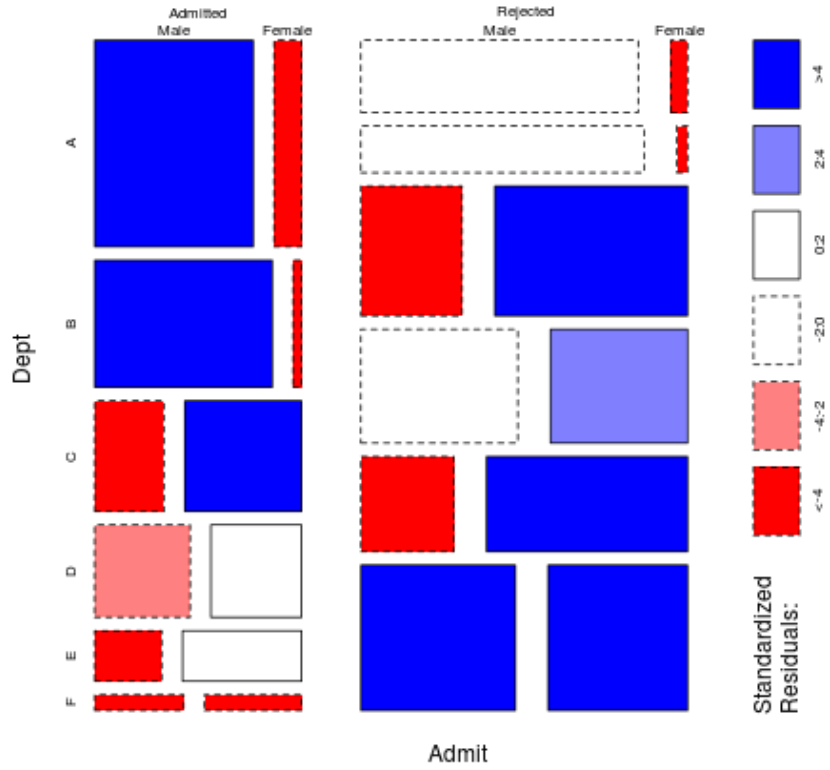


4.4. Mô hình hồi quy và Log-tuyến tính: Giảm dữ liệu tham số (Regression and Log-Linear Models: Parametric Data Reduction)

4.4.2. Mô hình log-tuyến tính (Log-linear model)

- Các mô hình log-tuyến tính gần đúng với phân bố xác suất đa chiều rời rạc (*discrete multidimensional probability distributions*).

- Cho một tập hợp các bộ dữ liệu có *n* chiều (*n* thuộc tính), có thể coi mỗi bộ dữ liệu là một điểm trong không gian *n* chiều. Các mô hình log-tuyến tính có thể được sử dụng để ước tính xác suất của từng điểm trong không gian đa chiều cho một tập hợp các thuộc tính rời rạc, dựa trên một tập hợp con nhỏ hơn của các tổ hợp chiều (*dimensional combinations*). Điều này cho phép không gian dữ liệu có chiều cao hơn (*higher-dimensional data space*) được xây dựng từ không gian có chiều thấp hơn (*lower-dimensional spaces*).



- Do đó, các mô hình log-tuyến tính cũng hữu ích cho việc giảm kích thước (vì các điểm có chiều thấp hơn thường chiếm ít không gian hơn các điểm dữ liệu ban đầu) và làm mịn dữ liệu (*data smoothing* - vì các ước tính tổng hợp trong không gian có chiều thấp hơn ít chịu các biến thể lấy mẫu hơn các ước tính trong không gian nhiều chiều hơn).

4.5. Biểu đồ (Histograms)

- Biểu đồ sử dụng tính năng tạo nhóm để phân phối dữ liệu gần đúng và là một hình thức giảm dữ liệu phổ biến.
- Biểu đồ cho thuộc tính A, phân vùng phân phối dữ liệu của A thành các tập con rời rạc, được gọi là buckets (nhóm) hoặc bins (thùng). Nếu mỗi nhóm:
  - Chỉ đại diện cho một cặp thuộc tính–giá trị/tần số duy nhất thì các nhóm đó được gọi là nhóm đơn.
  - Đại diện cho các phạm vi liên tục của thuộc tính đã cho được gọi là nhóm tổng hợp.

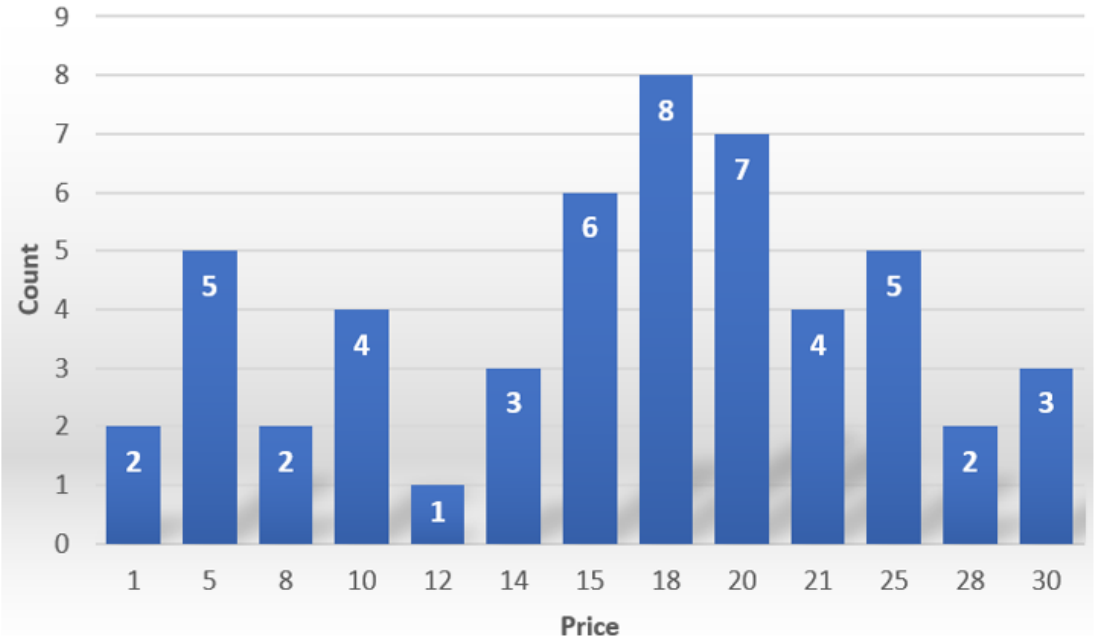
Price	Count
1	2
5	5
8	2
10	4
12	1
14	3
15	6
18	8
20	7
21	4
25	5
28	2
30	3

Price	Count
1-10	13
11-20	25
21-30	14

4.5. Biểu đồ (Histograms)

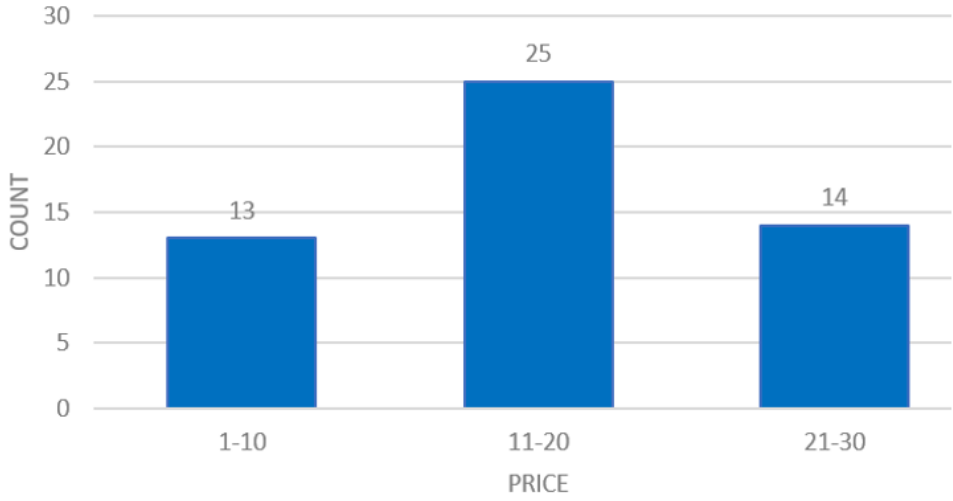
- ***Ví dụ***: Dữ liệu sau là danh sách giá bán đã được sắp xếp của các mặt hàng: 1, 1, 5, 5, 5, 5, 5, 8, 8, 10, 10, 10, 10, 12, 14, 14, 14, 15, 15, 15, 15, 15, 15, 18, 18, 18, 18, 18, 18, 18, 18, 20, 20, 20, 20, 20, 20, 20, 21, 21, 21, 21, 25, 25, 25, 25, 25, 28, 28, 30, 30, 30.

Price	Count
1	2
5	5
8	2
10	4
12	1
14	3
15	6
18	8
20	7
21	4
25	5
28	2
30	3



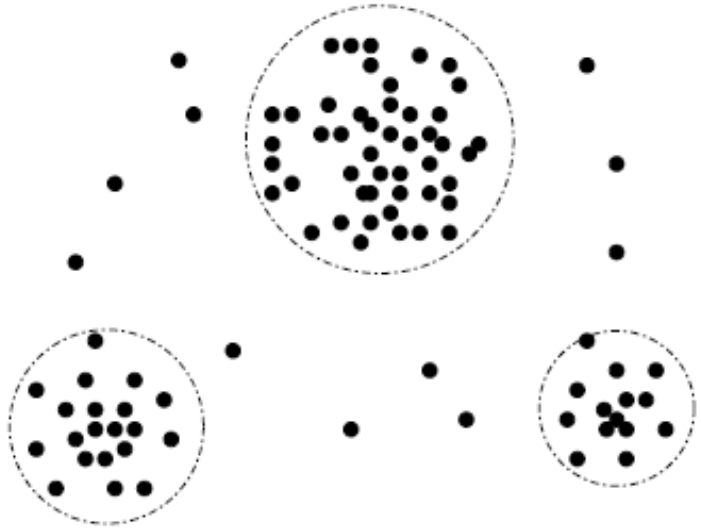
Biểu đồ sử dụng cho các nhóm giá đơn lẻ, mỗi nhóm đại diện cho một cặp giá- số lượng (hay tần suất)

Price	Count
1-10	13
11-20	25
21-30	14



Biểu đồ có chiều rộng bằng nhau cho giá, trong đó các giá trị được tổng hợp sao cho mỗi bucket có chiều rộng đồng đều là 10$

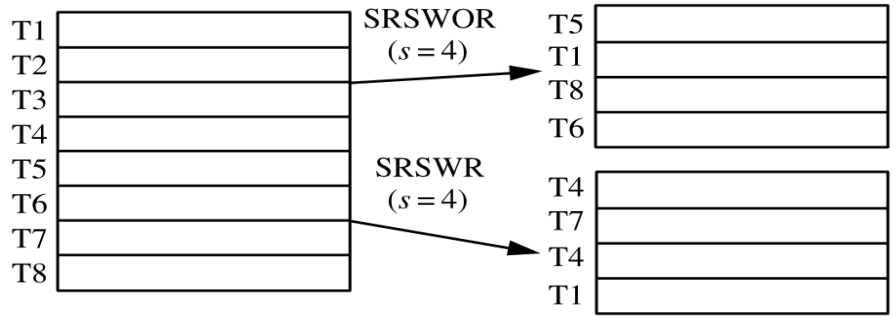
4.6. Phân cụm (Clustering)



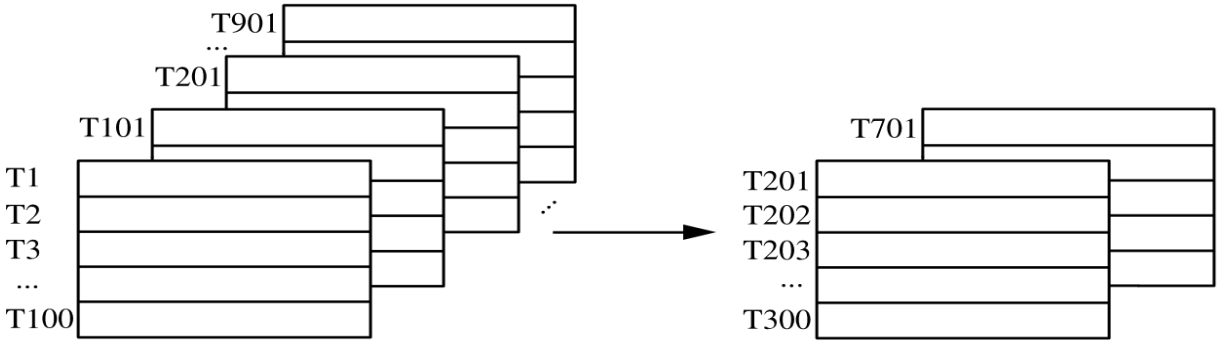
- Kỹ thuật phân cụm coi các bộ dữ liệu là đối tượng.
- Chúng phân chia các đối tượng thành các nhóm (*groups*) hoặc cụm (*cluster*), sao cho các đối tượng trong một cụm là "tương tự" (*similar*) với nhau và "không giống" (*dissimilar*) với các đối tượng trong các cụm khác.
- Sự tương đồng thường được định nghĩa theo mức độ "gần gũi" (*close*) của các vật thể trong không gian, dựa trên hàm khoảng cách (*distance function*). "Chất lượng" (*quality*) của một cụm có thể được biểu thị bằng đường kính (*diameter*) của nó, tức là khoảng cách tối đa giữa hai đối tượng bất kỳ trong cụm. Khoảng cách trung tâm (*centroid distance*) là một thước đo thay thế về chất lượng cụm và được định nghĩa là khoảng cách trung bình của từng đối tượng cụm từ tâm cụm (biểu thị "đối tượng trung bình" (*average object*) hoặc điểm trung bình (*point object*) trong không gian của cụm).

4.7. Lấy mẫu (Sampling)

- Lấy mẫu có thể được sử dụng như một kỹ thuật rút gọn dữ liệu vì nó cho phép một tập dữ liệu lớn được biểu diễn bằng một mẫu (hoặc tập hợp con) dữ liệu ngẫu nhiên nhỏ hơn nhiều.



Cluster sample
$(s = 2)$



Startified sample
(according to *age*)

T38	youth
T256	youth
T307	youth
T391	youth
T96	middle_aged
T117	middle_aged
T138	middle_aged
T263	middle_aged
T290	middle_aged
T308	middle_aged
T326	middle_aged
T387	middle_aged
T69	senior
T284	senior

T38	youth
T391	youth
T117	middle_aged
T138	middle_aged
T290	middle_aged
T326	middle_aged
T69	senior

4.7. Lấy mẫu (Sampling)

- Những cách lấy mẫu phổ biến:

  i. ***Mẫu ngẫu nhiên đơn giản không thay thế*** (*Simple random sample without replacement - SRSWOR*) : tất cả các bộ dữ liệu đều có khả năng được lấy mẫu như nhau.

  ii. ***Mẫu ngẫu nhiên đơn giản có thay thế*** (*Simple random sample with replacement - SRSWR*): sau khi một bộ được rút ra, nó sẽ được đặt lại vào D để có thể được rút lại.

4.7. Lấy mẫu (Sampling)

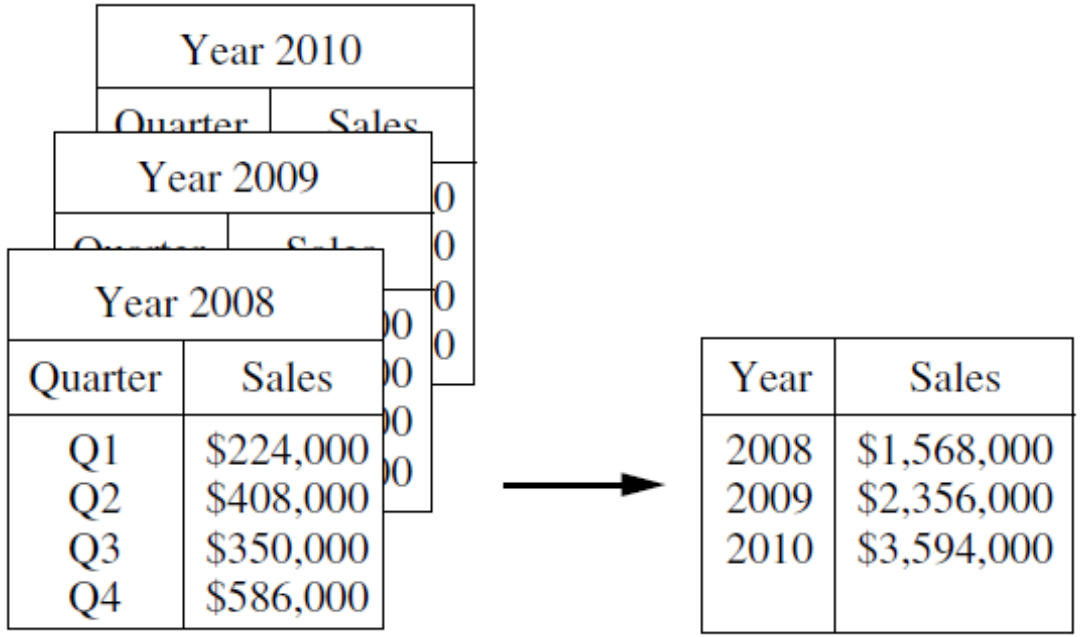
- Những cách lấy mẫu phổ biến:

iii. ***Mẫu cụm*** (*Cluster sample*): Nếu các bộ dữ liệu trong D được nhóm thành M "*cụm*" rời rạc lẫn nhau thì có thể thu được mẫu ngẫu nhiên đơn giản (*Simple random sample - SRS*) của ***s*** cụm, trong đó ***s < M***. Có thể áp dụng *SRSWOR* khi lấy mẫu của cụm.

iv. ***Mẫu phân tầng*** (*Stratifiedsample*): Nếu D được chia thành các phần rời rạc lẫn nhau gọi là tầng (*strata*), thì mẫu phân tầng (*stratified sample*) của D được tạo ra bằng cách lấy *SRS* ở mỗi tầng. Điều này giúp đảm bảo các tầng đều có mẫu đại diện, đặc biệt khi dữ liệu bị sai lệch.

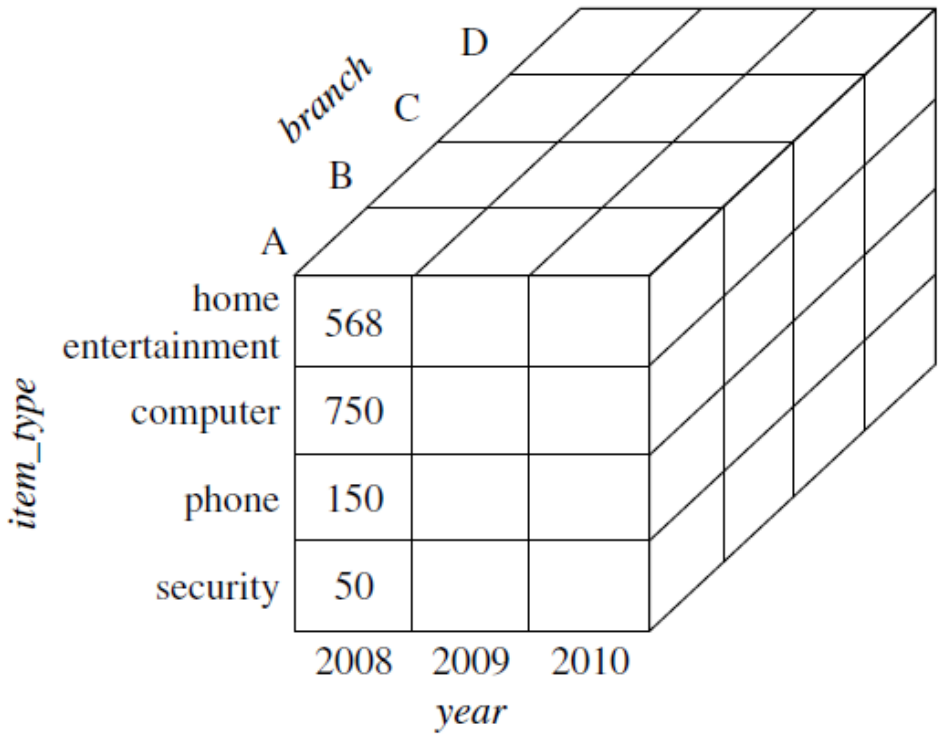
4.8. Tổng hợp khối dữ liệu (Data Cube Aggregation)

- Giả sử đã có số liệu tổng hợp doanh số của từng quý trong nhiều năm. Do chỉ quan tâm đến doanh số theo từng năm (tổng mỗi năm). Do đó, dữ liệu có thể được tổng hợp để dữ liệu thu được tóm tắt tổng doanh số bán hàng mỗi năm thay vì mỗi quý. Tập dữ liệu thu được có khối lượng nhỏ hơn, không làm mất thông tin cần thiết cho nhiệm vụ phân tích.



4.8. Tổng hợp khối dữ liệu (Data Cube Aggregation)

- ***Khối dữ liệu*** (*Data cubes*)
  - Khối dữ liệu lưu trữ thông tin tổng hợp đa chiều. Trong đó, mỗi ô chứa một giá trị dữ liệu tổng hợp, tương ứng với điểm dữ liệu trong không gian đa chiều.
  - Các khối dữ liệu cung cấp khả năng truy cập nhanh vào dữ liệu tóm tắt, được tính toán trước, từ đó mang lại lợi ích cho việc xử lý phân tích trực tuyến cũng như khai thác dữ liệu.



  - Khối được tạo ở mức trừu tượng thấp nhất được gọi là khối cơ sở (*base cuboid*) sao cho khối này phải có thể sử dụng được hoặc hữu ích cho việc phân tích.
  - Khối lập phương ở mức độ trừu tượng cao nhất là hình khối đỉnh, vì khối này cho dữ liệu tổng hợp (tổng doanh số, tổng chi phí, …).
  - Khi trả lời các yêu cầu khai thác dữ liệu, nên sử dụng hình khối nhỏ nhất có sẵn phù hợp với nhiệm vụ nhất định.

NỘI DUNG CHƯƠNG 3

1. Tổng quan về xử lý dữ liệu (*Data Preprocessing: An Overview*)
2. Làm sạch dữ liệu (*Data Cleaning*)
3. Tích hợp dữ liệu (*Data Integration*)
4. Giảm thiểu dữ liệu (*Data Reduction*)
5. Chuyển đổi dữ liệu và phân tách dữ liệu (*Data Transformation and Data Discretization*)
6. Thực hành tiền xử lý dữ liệu

5. CHUYỂN ĐỔI DỮ LIỆU VÀ PHÂN TÁCH DỮ LIỆU

(*Data Transformation and Data Discretization*)

5.1. Tổng quan về chiến lược chuyển đổi dữ liệu *(Data Transformation Strategies Overview)*

Các chiến lược chuyển đổi dữ liệu:

i. ***Làm mịn*** (*Smoothing*): có tác dụng loại bỏ nhiễu khỏi dữ liệu. Các kỹ thuật bao gồm *binning*, hồi quy (*regression*) và phân cụm (*regression*).

ii. ***Xây dựng thuộc tính*** (*Attribute construction* - hoặc xây dựng đối tượng - *feature construction*), trong đó các thuộc tính mới được xây dựng và bổ sung từ tập hợp các thuộc tính đã cho để hỗ trợ quá trình khai thác.

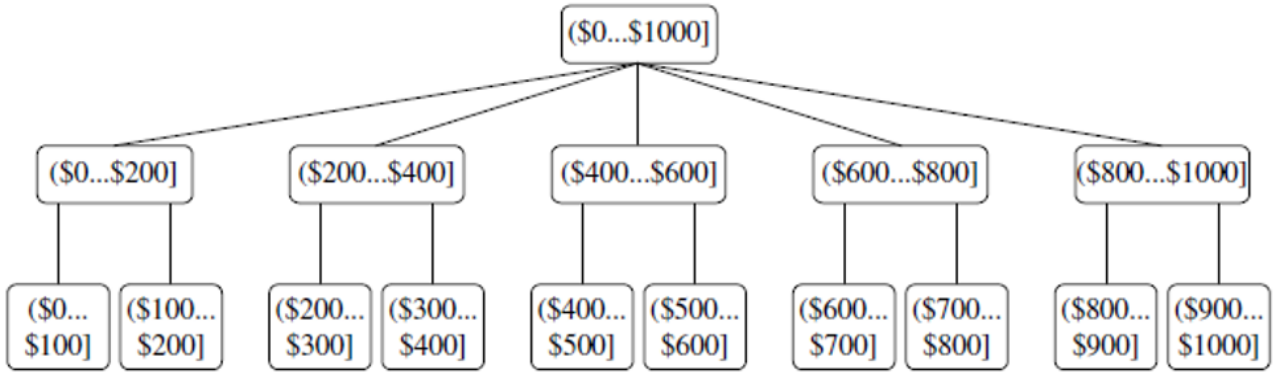
iii. ***Tổng hợp*** (*Aggregation*): trong đó các hoạt động tóm tắt (*summary*) hoặc tổng hợp (*aggregation*) được áp dụng cho dữ liệu. Ví dụ: dữ liệu bán hàng hàng ngày có thể được tổng hợp để tính tổng số tiền hàng tháng và hàng năm. Bước này thường được sử dụng trong việc xây dựng khối dữ liệu để phân tích dữ liệu ở nhiều mức độ trừu tượng (*abstraction levels*).

5.1. Tổng quan về chiến lược chuyển đổi dữ liệu *(Data Transformation Strategies Overview)*

Các chiến lược chuyển đổi dữ liệu bao gồm:

iv. ***Chuẩn hóa*** (*Normalization*): dữ liệu thuộc tính được chia tỷ lệ để nằm trong phạm vi nhỏ hơn, chẳng hạn như −1,0 đến 1,0 hoặc 0,0 đến 1,0.

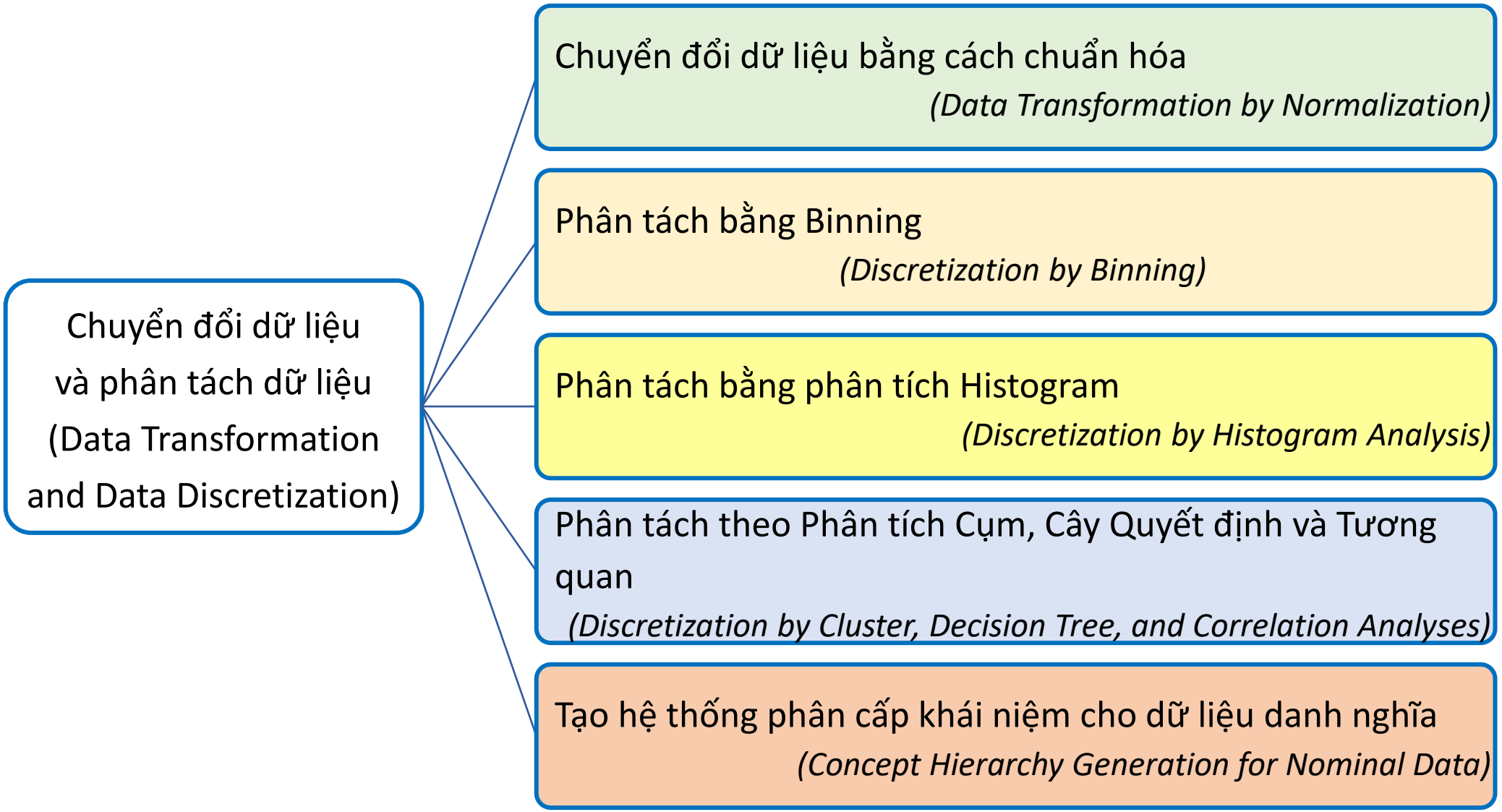
v. ***Rời rạc hóa*** (*Discretization*): các giá trị thô của thuộc tính số (ví dụ: tuổi) được thay thế bằng các nhãn khoảng (ví dụ: 0–10, 11–20, v.v...) hoặc các nhãn khái niệm (ví dụ: thanh niên, người lớn, người cao tuổi). Ngược lại, các nhãn có thể được tổ chức đệ quy thành các khái niệm cấp cao hơn, dẫn đến hệ thống phân cấp khái niệm cho thuộc tính số.



Hệ thống phân cấp khái niệm cho thuộc tính giá, trong đó một khoảng ($X ...$Y] biểu thị phạm vi từ $X đến $Y.

vi. ***Tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa*** *(Concept hierarchy generation for nominal data)*, trong đó các thuộc tính như đường phố có thể được khái quát hóa thành các khái niệm cấp cao hơn, như thành phố hoặc quốc gia. Nhiều hệ thống phân cấp cho các thuộc tính danh nghĩa được ẩn trong lược đồ cơ sở dữ liệu và có thể được xác định tự động ở cấp độ định nghĩa lược đồ.

Các phương pháp Chuyển đổi dữ liệu và phân tách dữ liệu



5.2. Chuyển đổi dữ liệu bằng cách chuẩn hóa (Data Transformation by Normalization)

- Đơn vị đo được sử dụng có thể ảnh hưởng đến việc phân tích dữ liệu.

  Ví dụ: thay đổi đơn vị đo từ mét sang inch cho chiều cao hoặc từ kilogam sang pound cho cân nặng có thể dẫn đến các kết quả rất khác nhau.
- Để tránh sự phụ thuộc vào việc lựa chọn đơn vị đo lường, dữ liệu phải được chuẩn hóa (*normalized*) hoặc tiêu chuẩn hóa (*standardized*). Điều này liên quan đến việc chuyển đổi dữ liệu để nằm trong phạm vi nhỏ hơn hoặc phổ biến hơn, chẳng hạn như [−1, 1] hoặc [0,0; 1,0].
- Nói chung, việc thể hiện một thuộc tính theo đơn vị nhỏ hơn sẽ dẫn đến phạm vi lớn hơn cho thuộc tính đó và do đó có xu hướng mang lại cho thuộc tính đó hiệu ứng (effect) hay "trọng lượng" (weight) lớn hơn.
- Việc chuẩn hóa dữ liệu cố gắng cung cấp cho tất cả các thuộc tính một trọng số bằng nhau (equal weight).

5.2. Chuyển đổi dữ liệu bằng cách chuẩn hóa (Data Transformation by Normalization)

- Lợi ích của việc chuẩn hóa đối với các thuật toán:
  - Đặc biệt hữu ích cho các thuật toán phân loại liên quan đến mạng neural (neural networks) hoặc đo khoảng cách như phân loại và phân cụm láng giềng gần nhất (nearest-neighbor classification and clustering).
  - Đối với thuật toán lan truyền ngược mạng neural (neural network backpropagation algorithm) để khai thác phân loại, việc chuẩn hóa các giá trị đầu vào cho từng thuộc tính được đo trong các bộ dữ liệu huấn luyện (training tuples) sẽ giúp tăng tốc giai đoạn học.
  - Đối với các phương pháp dựa trên khoảng cách (distance-based methods), việc chuẩn hóa giúp ngăn các thuộc tính có phạm vi lớn ban đầu (ví dụ: thu nhập) vượt trội hơn các thuộc tính có phạm vi nhỏ hơn ban đầu (ví dụ: thuộc tính nhị phân). Nó cũng hữu ích khi không có kiến thức trước về dữ liệu.

5.2. Chuyển đổi dữ liệu bằng cách chuẩn hóa (Data Transformation by Normalization)

- Các phương pháp để chuẩn hóa dữ liệu: cho A là một thuộc tính số có n giá trị quan sát, v_1, v_2, …, v_n.

  i. ***Chuẩn hóa tối thiểu-tối đa*** (*min-max normalization*) thực hiện chuyển đổi tuyến tính trên dữ liệu gốc. Giả sử min_A và max_A là các giá trị tối thiểu và tối đa của thuộc tính A. Chuẩn hóa tối thiểu - tối đa ánh xạ giá trị v_i của A đến v_{i0} trong phạm vi $[new\_min_A, new\_max_A]$ bằng cách tính toán

$$v_i' = \frac{v_i - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \qquad \textit{Công thức 8}$$

  - Chuẩn hóa tối thiểu tối đa bảo tồn các mối quan hệ giữa các giá trị dữ liệu ban đầu. Nó sẽ gặp lỗi "ngoài giới hạn" (*out-of-bounds*) sau này khi giá trị nhập vào nằm ngoài phạm vi dữ liệu ban đầu của A.
  - ***Ví dụ***: Giả sử giá trị tối thiểu và tối đa cho thuộc tính thu nhập (*income attribute*) lần lượt là 12.000$ và 98.000$. Ta ánh xạ thu nhập này vào phạm vi [0.0; 1.0]. Bằng cách chuẩn hóa tối thiểu tối đa, khi đó, giá trị 73,600$ cho thu nhập được chuyển thành:

$$\frac{73.600 - 12.000}{98.000 - 12.000}(1.0 - 0) + 0 = 0.716$$

5.2. Chuyển đổi dữ liệu bằng cách chuẩn hóa (Data Transformation by Normalization)

ii. ***Chuẩn hóa điểm z*** (*z-score normalization* hoặc chuẩn hóa trung bình bằng zero - *zero-mean normalization*): các giá trị cho thuộc tính A, được chuẩn hóa dựa trên giá trị trung bình (*mean* - tức là *average*) và độ lệch chuẩn (*standard deviation*) của ***A***. Một giá trị, v_i, của *A* được chuẩn hóa thành v_{i0} bằng cách tính toán.

$$v_i' = \frac{v_i - \bar{A}}{\sigma_A}$$ *Công thức 2-9*

trong đó

- $\bar{A}$ là giá trị trung bình của thuộc tính A. $\bar{A} = \frac{1}{n}(v_1 + v_2 + \cdots + v_n)$
- σ_A là độ lệch chuẩn của thuộc tính A. được tính là căn bậc hai của phương sai của A.
- Phương pháp chuẩn hóa này hữu ích khi không xác định mức tối thiểu và tối đa thực tế của thuộc tính A, hoặc khi có các ngoại lệ chi phối chuẩn hóa min-max.

Ví dụ: Giả sử giá trị trung bình và độ lệch chuẩn của thuộc tính thu nhập lần lượt là 54.000$ và 16.000$. Với chuẩn hóa điểm z, giá trị 73.600$ cho thu nhập được chuyển đổi thành *(73.600-54.000) /16.000=1.225*

5.2. Chuyển đổi dữ liệu bằng cách chuẩn hóa (Data Transformation by Normalization)

iii. ***Chuẩn hóa theo tỷ lệ thập phân*** (*normalization by decimal scaling*) thực hiện chuẩn hóa bằng cách di chuyển dấu thập phân của các giá trị thuộc tính *A*. Số dấu thập phân di chuyển phụ thuộc vào giá trị tuyệt đối tối đa của *A*. Một giá trị, v_i, của *A* được chuẩn hóa thành v_{i0} bằng cách tính toán

$$v_i' = \frac{v_i}{10^j}$$ *Công thức 12*

trong đó *j* là số nguyên nhỏ nhất sao cho $\max|v_i'|<1$.

Ví dụ: Giả sử rằng các giá trị được ghi lại của *A* nằm trong khoảng từ −986 đến 917. Giá trị tuyệt đối tối đa của *A* là 986. Để chuẩn hóa theo tỷ lệ thập phân, do đó chia mỗi giá trị cho 1000 (tức là $j = 3$) để −986 trở thành −0,986 và 917 thành 0,917.

5.2. Chuyển đổi dữ liệu bằng cách chuẩn hóa (Data Transformation by Normalization)

iii. ***Chuẩn hóa theo tỷ lệ thập phân*** (*normalization by decimal scaling*) thực hiện chuẩn hóa bằng cách di chuyển dấu thập phân của các giá trị thuộc tính *A*. Số dấu thập phân di chuyển phụ thuộc vào giá trị tuyệt đối tối đa của *A*. Một giá trị, v_i, của *A* được chuẩn hóa thành v_{i0} bằng cách tính toán

$$v_i' = \frac{v_i}{10^j}$$ *Công thức 12*

trong đó *j* là số nguyên nhỏ nhất sao cho $\max|v_i'|<1$.

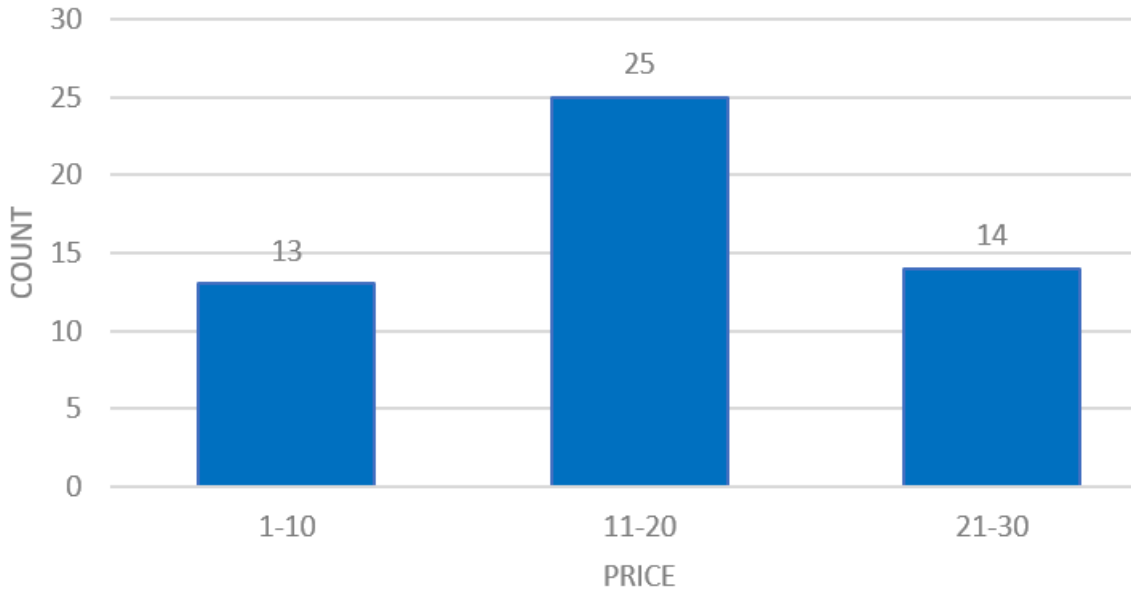
Ví dụ: Giả sử rằng các giá trị được ghi lại của *A* nằm trong khoảng từ −986 đến 917. Giá trị tuyệt đối tối đa của *A* là 986. Để chuẩn hóa theo tỷ lệ thập phân, do đó chia mỗi giá trị cho 1000 (tức là $j = 3$) để −986 trở thành −0,986 và 917 thành 0,917.

5.3. Phân tách bằng Binning (Discretization by Binning)

- Binning là một kỹ thuật tách từ trên xuống dựa trên một số bins (thùng) được chỉ định.
- Phương pháp này cũng được sử dụng để giảm dữ liệu và tạo hệ thống phân cấp khái niệm.
- Ví dụ, các giá trị thuộc tính có thể được phân tách bằng cách áp dụng tính năng phân chia bằng chiều rộng hoặc tần suất bằng nhau, và sau đó thay thế mỗi giá trị bin bằng giá trị trung bình (mean) hoặc trung vị (median) của bin. Các kỹ thuật này có thể được áp dụng đệ quy cho các phân vùng kết quả (resulting partitions) để tạo hệ thống phân cấp khái niệm (concept hierarchies).
- Binning không sử dụng thông tin lớp và do đó là một kỹ thuật phân tách không được giám sát (unsupervised discretization technique). Nó nhạy cảm với số lượng bins do người dùng chỉ định, cũng như sự hiện diện của các ngoại lệ.

5.5. Phân tách bằng phân tích Histogram *(Discretization by Histogram Analysis)*

- Giống như *binning*, phân tích biểu đồ là một kỹ thuật phân tách không giám sát vì nó không sử dụng thông tin lớp.
- Biểu đồ phân chia các giá trị của một thuộc tính, A, thành các phạm vi rời rạc được gọi là *buckets* (xô) hoặc *bins* (thùng). trong biểu đồ có chiều rộng bằng nhau (*equal-width histogram*), các giá trị được phân vùng thành các phân vùng hoặc phạm vi có kích thước bằng nhau.



5.6. Phân tách theo Phân tích Cụm, Cây Quyết định và Tương quan

(Discretization by Cluster, Decision Tree, and Correlation Analyses)

– ***Phân tích cụm*** (*Cluster analysis*):

- Thuật toán phân cụm có thể được áp dụng để phân tách một thuộc tính số, ***A***, bằng cách phân vùng các giá trị của ***A*** thành các cụm (*clusters*) hoặc nhóm (*groups*).
- Việc phân cụm xem xét sự phân bố của ***A***, cũng như sự gần gũi của các điểm dữ liệu, và do đó có thể tạo ra kết quả phân tách chất lượng cao.
- Phân cụm có thể được sử dụng để tạo hệ thống phân cấp khái niệm cho A bằng cách tuân theo chiến lược chia tách từ trên xuống (*top-down splitting*) hoặc chiến lược hợp nhất từ dưới lên (*bottom-up merging*), trong đó mỗi cụm tạo thành một nút của hệ thống phân cấp khái niệm.
- Mỗi cụm hoặc phân vùng ban đầu có thể được phân tách thêm thành nhiều cụm con, tạo thành một cấp thấp hơn của hệ thống phân cấp. Sau này, các cụm được hình thành bằng cách liên tục nhóm các cụm lân cận để hình thành các khái niệm cấp cao hơn.

5.6. Phân tách theo Phân tích Cụm, Cây Quyết định và Tương quan

(Discretization by Cluster, Decision Tree, and Correlation Analyses)

– ***Kỹ thuật tạo cây quyết định để phân loại*** (*generate decision trees for classification*)

- Sử dụng cách tiếp cận chia tách từ trên xuống (*top-down splitting approach*).
- Phương pháp tiếp cận cây quyết định có sử dụng thông tin nhãn lớp.
- *Entropy* là thước đo được sử dụng phổ biến nhất cho mục đích phân phối lớp. Để phân tách một thuộc tính số, A, phương pháp chọn giá trị của A có *entropy* tối thiểu (*minimum entropy*) làm điểm tách (*split-point*) và phân vùng đệ quy (*recursively partitions*) các khoảng kết quả để đi đến sự phân tách phân cấp (*hierarchical discretization*).
- Sự riêng biệt như vậy tạo thành một hệ thống phân cấp khái niệm (*concept hierarchy*) cho A.
- Ví dụ: có thể có một tập dữ liệu về các triệu chứng của bệnh nhân (các thuộc tính) trong đó mỗi bệnh nhân có nhãn lớp chẩn đoán liên quan. Thông tin phân phối lớp được sử dụng trong tính toán và xác định các điểm phân chia (giá trị dữ liệu để phân vùng phạm vi thuộc tính). Theo trực giác, ý tưởng chính là chọn các điểm chia sao cho một phân vùng kết quả nhất định chứa càng nhiều tuple của cùng một lớp càng tốt.

5.6. Phân tách theo Phân tích Cụm, Cây Quyết định và Tương quan

(Discretization by Cluster, Decision Tree, and Correlation Analyses)

– ***Thước đo tương quan*** (*measures of correlation*)

- Về cơ bản, để phân tách chính xác, các tần suất lớp tương đối phải khá nhất quán trong một khoảng thời gian. Do đó, nếu hai khoảng liền kề có phân phối các lớp rất giống nhau, thì các khoảng có thể được hợp nhất. Nếu không, chúng vẫn tách biệt.
- *ChiMerge* là một phương pháp phân tách dựa trên χ^2.
- Các phương pháp phân tách đã nghiên cứu cho đến thời điểm này đều sử dụng một chiến lược chia tách từ trên xuống. Điều này trái ngược với *ChimerGE*, sử dụng cách tiếp cận từ dưới lên bằng cách tìm các khoảng lân cận tốt nhất và sau đó hợp nhất chúng bằng cách đệ quy để tạo thành các khoảng lớn hơn.
- Cũng như phân tích cây quyết định, *ChiMerge* được giám sát ở chỗ nó sử dụng thông tin lớp.

5.7. Tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa

(Concept Hierarchy Generation for Nominal Data)

Có bốn phương pháp để tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa:

i. ***Đặc tả thứ tự của các thuộc tính một cách rõ ràng ở cấp độ lược đồ bởi người dùng hoặc chuyên gia***:

- Hệ thống phân cấp khái niệm cho các thuộc tính hoặc kích thước danh nghĩa thường liên quan đến một nhóm thuộc tính.
- Người dùng hoặc chuyên gia có thể dễ dàng xác định một hệ thống phân cấp khái niệm bằng cách chỉ định thứ tự một phần hoặc toàn bộ của các thuộc tính ở cấp lược đồ. Ví dụ: giả sử cơ sở dữ liệu quan hệ chứa nhóm thuộc tính sau: đường phố, thành phố, tỉnh hoặc tiểu bang và quốc gia.
- Một hệ thống phân cấp có thể được xác định bằng cách chỉ định thứ tự giữa các thuộc tính này ở cấp lược đồ như: ***đường phố < thành phố < tỉnh hoặc tiểu bang < quốc gia***.

5.7. Tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa (Concept Hierarchy Generation for Nominal Data)

ii. *Đặc tả một phần của hệ thống phân cấp bằng cách nhóm dữ liệu rõ ràng*:

- Về cơ bản là định nghĩa thủ công của một phần của hệ thống phân cấp khái niệm.
- Trong một cơ sở dữ liệu lớn, việc xác định toàn bộ hệ thống phân cấp khái niệm bằng cách liệt kê giá trị rõ ràng là không thực tế. Ngược lại, chúng ta có thể dễ dàng chỉ định các nhóm rõ ràng cho một phần nhỏ dữ liệu cấp trung gian.
- Ví dụ, sau khi chỉ định rằng tỉnh và quốc gia tạo thành một hệ thống phân cấp ở cấp lược đồ, người dùng có thể xác định một số cấp trung gian theo cách thủ công, chẳng hạn như "{Kon Tum, Gia Lai, Đắk Lắk, Đắk Nông, Lâm Đồng } ⊂ Tây nguyên" và "Bắc Trung Bộ, Nam Trung Bộ, Tây nguyên" ⊂ Miền Trung

5.7. Tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa (Concept Hierarchy Generation for Nominal Data)

iii. *Đặc điểm kỹ thuật của một tập hợp các thuộc tính, nhưng không phải thứ tự một phần của chúng* (*Specification of a set of attributes, but not of their partial ordering*):

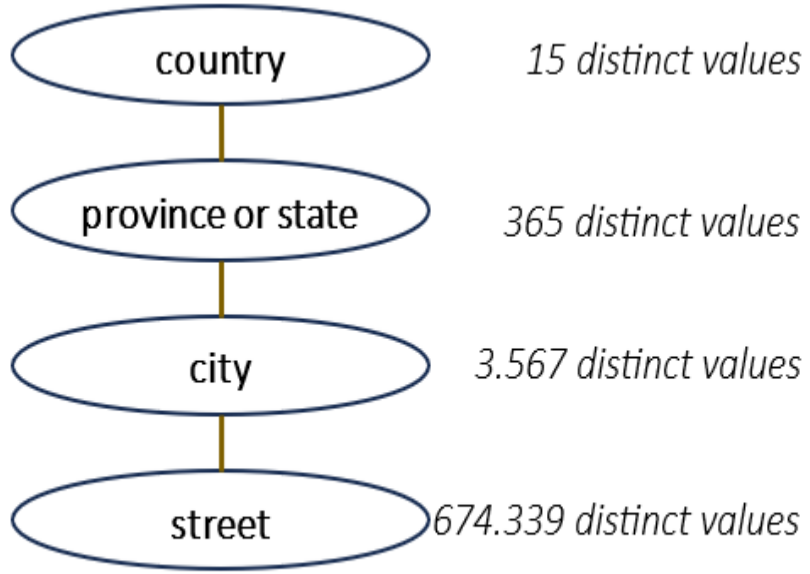
Người dùng có thể chỉ định một tập hợp các thuộc tính tạo thành một hệ thống phân cấp khái niệm, nhưng bỏ qua để nêu rõ thứ tự một phần của chúng. Sau đó, hệ thống có thể cố gắng tự động tạo thứ tự thuộc tính để xây dựng một hệ thống phân cấp khái niệm có ý nghĩa.

5.7. Tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa (Concept Hierarchy Generation for Nominal Data)

iii. Đặc điểm kỹ thuật của một tập hợp các thuộc tính, nhưng không phải thứ tự một phần của chúng

Ví dụ: Giả sử người dùng chọn một tập hợp các thuộc tính như đường-phố, quốc gia, tỉnh hoặc tiểu bang và thành phố từ cơ sở dữ liệu sẵn có, nhưng không chỉ định thứ tự phân cấp giữa các thuộc tính.

- ***B1***: Sắp xếp các thuộc tính theo thứ tự tăng dần dựa trên số lượng giá trị riêng biệt trong mỗi thuộc tính. Điều này dẫn đến như sau (trong đó số giá trị riêng biệt cho mỗi thuộc tính được hiển thị trong ngoặc đơn): quốc gia (15), tỉnh hoặc tiểu bang (365), thành phố (3.567) và đường phố (674.339).
- ***B2***: tạo hệ thống phân cấp từ trên xuống theo thứ tự sắp xếp, với thuộc tính đầu tiên ở cấp trên cùng và thuộc tính cuối cùng ở cấp dưới cùng.



Tự động tạo hệ thống phân cấp khái niệm lược đồ dựa trên số lượng giá trị thuộc tính riêng biệt.

- ***B3***: có thể kiểm tra hệ thống phân cấp được tạo và khi cần thiết, sửa đổi nó để phản ánh các mối quan hệ ngữ nghĩa mong muốn giữa các thuộc tính.

Trong ví dụ này, rõ ràng là không cần phải sửa đổi hệ thống phân cấp đã tạo.

5.7. Tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa (Concept Hierarchy Generation for Nominal Data)

iv. Đặc tả chỉ một phần của bộ thuộc tính (*Specification of only a partial set of attributes*):

- Đôi khi người dùng có thể bất cẩn khi xác định một hệ thống phân cấp hoặc chỉ có một ý tưởng mơ hồ về những gì nên được đưa vào hệ thống phân cấp. Do đó, người dùng có thể chỉ bao gồm một tập hợp nhỏ của các thuộc tính có liên quan trong đặc tả phân cấp.

  Ví dụ: thay vì bao gồm tất cả các thuộc tính có liên quan theo thứ bậc cho vị trí, người dùng có thể chỉ định đường phố và thành phố.

- Để xử lý các hệ thống phân cấp được chỉ định một phần như vậy, điều quan trọng là phải nhúng ngữ nghĩa dữ liệu vào lược đồ cơ sở dữ liệu để các thuộc tính có kết nối ngữ nghĩa chặt chẽ có thể được ghim lại với nhau. Theo cách này, đặc tả của một thuộc tính có thể kích hoạt cả một nhóm các thuộc tính được liên kết chặt chẽ về mặt ngữ nghĩa bị "kéo vào" để tạo thành một hệ thống phân cấp hoàn chỉnh.

5.7. Tạo hệ thống phân cấp khái niệm cho dữ liệu danh nghĩa (Concept Hierarchy Generation for Nominal Data)

iv. Đặc tả chỉ một phần của bộ thuộc tính

Ví dụ: Giả sử rằng một chuyên gia khai thác dữ liệu đã ghim năm thuộc tính số, đường phố, thành phố, tỉnh hoặc tiểu bang và quốc gia, bởi vì chúng được liên kết chặt chẽ về mặt ngữ nghĩa liên quan đến khái niệm vị trí. Nếu người dùng chỉ định thuộc tính thành phố cho vị trí xác định thứ bậc, hệ thống có thể tự động kéo tất cả năm thuộc tính có liên quan về ngữ nghĩa để tạo thành một hệ thống phân cấp. Người dùng có thể chọn loại bỏ bất kỳ thuộc tính nào trong số này (ví dụ: số và đường phố) khỏi hệ thống phân cấp, giữ thành phố là cấp khái niệm thấp nhất.

NỘI DUNG CHƯƠNG 3

1. Tổng quan về xử lý dữ liệu (*Data Preprocessing: An Overview*)
2. Làm sạch dữ liệu (*Data Cleaning*)
3. Tích hợp dữ liệu (*Data Integration*)
4. Giảm thiểu dữ liệu (*Data Reduction*)
5. Chuyển đổi dữ liệu và phân tách dữ liệu (*Data Transformation and Data Discretization*)
6. Thực hành tiền xử lý dữ liệu

6. THỰC HÀNH

6.1. Data Scientist Job Market in the U.S.

- Bối cảnh

- Báo cáo lực lượng lao động LinkedIn tháng 8 năm 2018 tại Hoa Kỳ cho biết đang thiếu 151.717 người có kỹ năng về khoa học dữ liệu, đặc biệt là tình trạng thiếu hụt trầm trọng ở New York. Thành phố York, Khu vực Vịnh San Francisco và Los Angeles.
- Để giúp những người tìm việc hiểu rõ hơn về thị trường việc làm, dựa trên thông tin trên trang web Indeed và thu thập thông tin về gần 7.000 việc làm data scientist trên khắp Hoa Kỳ.

- Nội dung dữ liệu

- Gồm 15 file, mỗi file chứa thông tin về các công ty thuộc thành phố hoặc khu vực cụ thể.
- Cấu trúc các table

Field name	*Meaning*	*Data type*
position	Position Title	text
company	Company Name	text
description	Job Description	text
reviews	Number of Reviews for the Company	text
location	Location of the Job	text

6.1. Data Scientist Job Market in the U.S.

- ***Các câu hỏi có thể được nêu ra đối với nguồn dữ liệu này***
  - Ai được thuê? Nhà tuyển dụng mong muốn loại tài năng nào khi thuê một nhà khoa học dữ liệu?
  - Vị trí nào có nhiều cơ hội nhất?
  - Những kỹ năng, công cụ, bằng cấp hoặc chuyên ngành nào mà nhà tuyển dụng mong muốn nhất đối với các nhà khoa học dữ liệu?
  - Sự khác biệt giữa nhà khoa học dữ liệu, kỹ sư dữ liệu và nhà phân tích dữ liệu là gì?
  - Có thể phát triển một thuật toán phân loại hiệu quả để phân biệt các loại công việc có trong dữ liệu không?

6.1. Data Scientist Job Market in the U.S.

- *Các yêu cầu sinh viên cần thực hiện*

- Tích hợp 15 file dữ liệu do GV cung cấp thành 1 file dữ liệu duy nhất (với tên file đề nghị là 'DS_JobMarketInUS'.
- Kiểm tra kiểu dữ liệu, miền giá trị của tất cả các thuộc tính.
- Xác định các giá trị bị trống, không hợp lệ (nếu có). Từ đó xác định các công việc cần làm khi tiền xử lý dữ liệu.
- Xử lý dữ liệu trong field *reviews*. Giả sử có quy ước:
  - Giá trị null: nếu cùng 1 công ty nhưng những chức danh (position) khác có lượt reviews thì lấy số lượng reviews trung bình của các chức danh khác làm số reviews cho chức danh đang bị null, nếu không thể lấy giá trị trung bình (VD: công ty chỉ có duy nhất 1 chức danh này) thì sẽ nhận giá trị là 0
  - Chỉ giữ lại giá trị số và chuyển kiểu dữ liệu hiện tại sang kiểu dữ liệu số nguyên.

6.2. Marketing Analytics

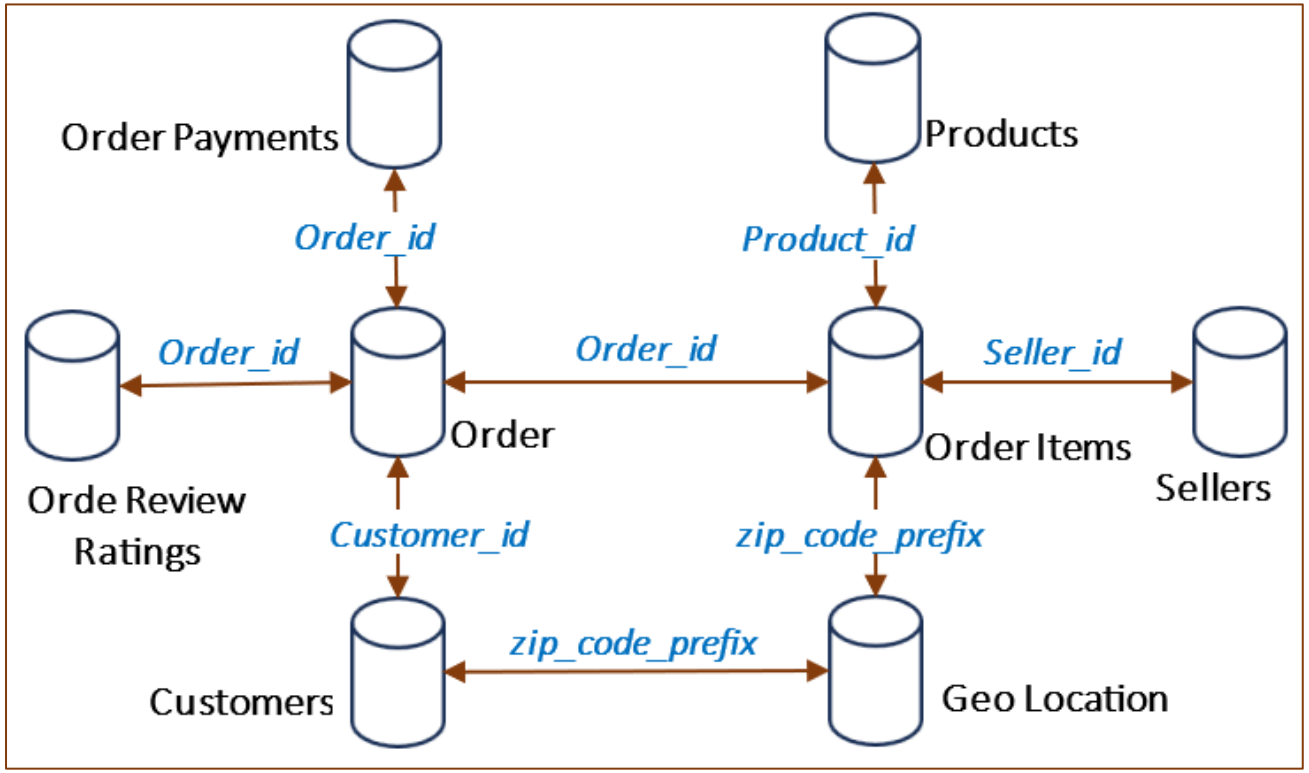
- Giới thiệu về tập dữ liệu

- Bối cảnh
  - Khách hàng muốn hợp tác với doanh nghiệp ABC tại Ấn Độ. Vì vậy, khách muốn được trợ giúp trong việc đo lường, quản lý và phân tích hiệu quả kinh doanh của doanh nghiệp ABC.
  - ABC thuê bạn làm nhà phân tích cho dự án này, nơi khách hàng yêu cầu cung cấp dữ liệu thúc đẩy những hiểu biết về kinh doanh, về hành vi của khách hàng, người bán, sản phẩm và về kênh phân phối, v.v...
  - Trong khi thực hiện dự án này, bạn phải làm sạch dữ liệu (nếu cần) trước khi phân tích nó.

6.2. Marketing Analytics

- Giới thiệu về tập dữ liệu

- Các bảng dữ liệu
  - *Customers*: Thông tin khách hàng.
  - *Geo_Location*: Chi tiết vị trí.
  - *Sellers*: Thông tin người bán.
  - *Products*: Thông tin sản phẩm.
  - *Orders*: Thông tin đơn hàng như số đơn đặt hàng, id sản phẩm, trạng thái, ngày đặt hàng, v.v.
  - *Order_Items*: Thông tin cấp độ đơn hàng.
  - *Order_Payments*: Thông tin thanh toán đơn hàng.
  - *Order_Review_Ratings*: Đánh giá của khách hàng ở cấp độ đơn hàng.



6.2. Marketing Analytics

- ***Một trong số mẫu câu hỏi cần giải quyết trong khuôn khổ phân tích này***
  - *Thực hiện phân tích thăm dò chi tiết*

a. Xác định và tính toán các số liệu như: Tổng doanh thu, Tổng số lượng, số lượng sản phẩm, số lượng danh mục sản phẩm, số lượng người bán, số lượng địa điểm, số lượng kênh (channels), số lượng phương thức thanh toán v.v...

b. Mỗi tháng có thêm được bao nhiêu khách hàng mới?

c. Hiểu rõ việc giữ chân khách hàng theo từng tháng

d. Doanh thu từ khách hàng hiện tại/mới như thế nào theo từng tháng

e. Hiểu xu hướng/tính thời vụ của doanh số bán hàng, số lượng theo danh mục, địa điểm, tháng, tuần, ngày, giờ, kênh, phương thức thanh toán, v.v...

f. Tên các sản phẩm phổ biến theo từng tháng, người bán, tiểu bang, danh mục.

g. Tên các danh mục phổ biến theo tiểu bang, tháng

h. Danh sách top 10 sản phẩm đắt nhất được sắp xếp theo giá.

6.2. Marketing Analytics

- ***Một trong số mẫu câu hỏi cần giải quyết trong khuôn khổ phân tích này***
  - *Thực hiện phân khúc khách hàng/người bán*

    a. Chia khách hàng thành các nhóm dựa trên doanh thu tạo ra

    b. Chia người bán thành các nhóm dựa trên doanh thu tạo ra

  - *Những sản phẩm nào được bán cùng nhau*

    Gợi ý: Chúng ta cần tìm ra sản phẩm nào trong số 10 sản phẩm kết hợp hàng đầu đang được bán cùng nhau ở mỗi giao dịch. (kết hợp 2 hoặc 3 sản phẩm được mua cùng nhau)

  - *Hành vi thanh toán*

    a. Khách hàng đang thanh toán như thế nào?

    b. Kênh thanh toán nào được nhiều khách hàng sử dụng nhất?

  - *Sự hài lòng của khách hàng đối với chủng loại & sản phẩm*

    a. Những danh mục nào (top 10) được xếp hạng tối đa và xếp hạng tối thiểu?

    b. Những sản phẩm nào (top10) được xếp hạng tối đa và xếp hạng tối thiểu?

    c. Đánh giá trung bình theo vị trí, người bán, sản phẩm, danh mục, tháng, v.v.

6.2. Marketing Analytics

- *Các yêu cầu sinh viên cần thực hiện*

- Kiểm tra kiểu dữ liệu, miền giá trị của tất cả các thuộc tính.
- Xác định các giá trị bị trống, không hợp lệ (nếu có). Từ đó xác định các công việc cần làm khi tiền xử lý dữ liệu.
- Để dữ liệu dễ hiểu, gọn nhẹ. Chuyển các field về ID thành các giá trị chuỗi số gồm 5 ký số (đánh số từ "00001" trở đi), Sau đó, thêm vào trước dãy số đó:
  - Chuỗi "Cus" nếu là CustomID
  - Chuỗi "Ord" nếu là OrderID
  - Chuỗi "Pro" nếu là ProID
  - …

6.2. Marketing Analytics

- *Các yêu cầu sinh viên cần thực hiện*
  - Để khai thác các tập phổ biến, một đề xuất là chuyển dạng thể hiện của dữ liệu, ví dụ:

Dữ liệu gốc

OrderID	ProductID	Quantity	...
1	X	4	
	Y	2	
	Z	7	
2	W	1	
	Z	8	
3	Y	9	
...	...		

⇨

Dữ liệu sau khi chuyển đổi

OrderID	X	Y	Z	W
1	1	1	1	0
2	0	0	1	1
3	0	1	0	0
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

Hoặc

⇨

Dữ liệu sau khi chuyển đổi

OrderID	X	Y	Z	W
1	4	2	7	0
2	0	0	8	1
3	0	9	0	0
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

6.3. *Chuyển đổi dạng thể hiện của dữ liệu*

a. Giả sử cho trước dữ liệu như bảng bên trái, sử dụng kiến thức về SQL để chuyển đổi sang dữ liệu như hình bên phải.

b. Ngược lại, giả sử cho trước dữ liệu như bảng bên phải, sử dụng kiến thức về SQL để chuyển đổi sang dữ liệu như hình bên trái.

nationality	*male*	*female*
USA	264	303
CHN	153	251
AUS	211	220
BRA	269	216
GER	237	204
...	...	...

nationality	quantity	sex
USA	264	male
USA	303	female
CHN	153	male
CHN	251	female
AUS	211	male
AUS	220	female
BRA	269	male
BRA	216	female
...	...	...


Q&A
How
Why
When
Where
What
Who