

# **Phát triển**

## **Các hệ thống thông minh**

# Chương 2

## Quá trình khoa học dữ liệu

Ứng dụng các kiến thức về khoa học dữ liệu được thực hiện theo từng bước. Có thể gọi các bước đó là một quá trình. Liên quan đến quá trình khoa học dữ liệu, chương này trình bày về quá trình thăm dò dữ liệu, rồi quá trình khoa học dữ liệu. Một số thí dụ minh họa được thể hiện bằng ngôn ngữ Python.

### 2.1. Quá trình thăm dò dữ liệu

Sau đây đề cập các khái niệm về hệ thống thông tin, hệ thống trợ giúp quyết định DSS<sup>1</sup> và việc sử dụng dữ liệu trong các hệ thống đó. Việc *thăm dò*<sup>2</sup> dữ liệu được xem như một quá trình, được thực hiện như công nghệ<sup>3</sup> về dữ liệu.

Nội dung về quá trình thăm dò dữ liệu gồm:

- *Quá trình thăm dò dữ liệu*, với các bước<sup>4</sup> (i) thăm dò không gian vấn đề<sup>5</sup>; (ii) thăm dò không gian lời giải<sup>6</sup>; (iii) xác định phương pháp thực hiện<sup>7</sup>; (iv) khai phá dữ liệu. Việc thăm dò là việc khai phá và sử dụng mô hình;
- *Các công cụ khai phá dữ liệu, mô hình hóa dữ liệu*. Ở đây có các qui tắc vàng, các kiểu<sup>8</sup> mô hình (i) mô hình tĩnh, động; (ii) mô hình dự báo và giải thích<sup>9</sup>; (iii) mô hình tĩnh và học liên tục.

Dữ liệu cần cho quá trình khai phá dữ liệu như sản phẩm cần cho thị trường. Người ta có thể lấy thí dụ về tầm quan trọng của sản phẩm đối với thị trường, để hiểu rõ về vai trò của dữ liệu đối với quá trình xử lý thống kê.

Trong thời gian này, khoa học về thống kê được quan tâm, bởi lẽ trong hoàn cảnh có nhiều dữ liệu, các quá trình xử lý dữ liệu, ra quyết định... đều cần đến thống kê. Khai phá dữ liệu cũng được xem như quá trình phân tích thống kê.

Liên quan đến hệ trợ giúp quyết định, có nhiều định nghĩa. Định nghĩa ban đầu của DSS là *một hệ thống chuyên trợ giúp cho những người quản lý trong công việc nửa cấu trúc*. DSS là trợ lý cho các chuyên gia ra quyết định để tăng năng lực

<sup>1</sup> DSS: hệ thống trợ giúp quyết định, Decision Support Systems sử dụng thuật ngữ trợ giúp, thay vì trợ giúp.

<sup>2</sup> Phân biệt Mining: *khai phá*; Exploration: *thăm dò*; Discovery: *khám phá*.

<sup>3</sup> Phân biệt công nghệ với kĩ thuật, thuật toán, phương pháp.

<sup>4</sup> Stage: bước, trong quá trình.

<sup>5</sup> Problem: vấn đề, bài toán. Từ sau sẽ sử dụng thuật ngữ vấn đề.

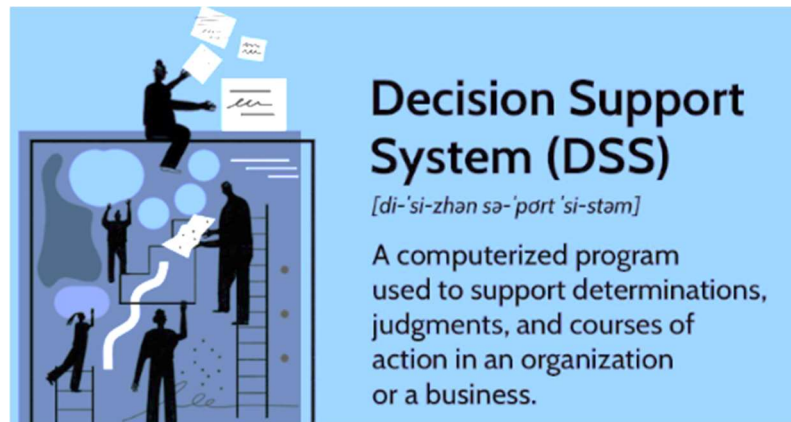
<sup>6</sup> Solution: sử dụng thuật ngữ lời giải, thay vì nghiệm; hoặc sử dụng thuật ngữ giải pháp.

<sup>7</sup> Implementaion: thực hiện, thay vì cài đặt.

<sup>8</sup> Type: kiểu; Kind: dạng.

<sup>9</sup> Explanatory: giải thích.

của họ, nhưng không được thay thế việc quyết định. Mục tiêu của họ là phải có cách giải quyết ở những nơi cần sự xét đoán hoặc giải quyết những cái không thể trợ giúp hoàn toàn bởi thuật toán. Tuy không hoàn toàn đồng tình với định nghĩa này thì người ta có thể hiểu được như vậy trong định nghĩa ban đầu đó rằng *nó có thể là một hệ thống dựa trên máy tính*, rằng nó hoạt động trực tuyến và tuyệt vời hơn nếu được trang bị đồ họa. Định nghĩa trên đã mở ra nhiều ý khác. Vài định nghĩa khác sinh ra sự *bất đồng* đáng kể khi xác định DSS là gì. Một số người hoài nghi thậm chí cho rằng DSS chỉ là *đồ án*. Trong phân tích dữ liệu lớn, người ta gắn kiến thức về kho dữ liệu với quá trình ra quyết định. Sau đây là định nghĩa DSS của Little, năm 1970.



**Hình. Hệ trợ giúp quyết định**

*Định nghĩa: Hệ trợ giúp quyết định DSS là tập hợp cơ sở mô hình của các thủ tục cho quá trình xử lý dữ liệu và phán đoán để trợ giúp người quản lý trong công việc tạo các phương án của họ.*

Bonczek, năm 1980, định nghĩa DSS như một hệ thống dựa trên cơ sở máy tính phù hợp với ba thành phần thích hợp (i) *hệ ngôn ngữ*, ứng với chiếc máy cho phép truyền thông giữa người sử dụng và các thành phần khác của DSS; (ii) *hệ thống thông tin*, nơi lưu các vấn đề đã biết trong DSS tức là dữ liệu hoặc các thủ tục; (iii) *hệ xử lý lỗi*, cho phép liên kết giữa hai thành phần, bao gồm một hoặc nhiều khả năng xử lý lỗi cần thiết cho việc tạo phương án giải quyết.

Trong quá trình ra quyết định, nhà quản lý sẽ thăm dò dữ liệu. Công việc này được thực hiện như một quá trình. Quá trình này bảo đảm tất cả những bên tham gia được huấn luyện, tự nguyện, có kì vọng<sup>1</sup> phù hợp, và bảo đảm thu được nhiều lợi theo công sức đầu vào. Quá trình này là quá trình thăm dò dữ liệu.

Trong tài liệu về *khai phá dữ liệu*, việc chuẩn bị dữ liệu liên quan đến các nhiệm vụ (i) làm sạch dữ liệu; (ii) quản lý dữ liệu thiếu; (iii) xác định việc phân lớp sai; (iv) các phương pháp đồ họa cho phép xác định dữ liệu lẻ; (v) đo độ chụm và độ tản

<sup>1</sup> Expectation: kì vọng.

mát; (vi) chuyển đổi dữ liệu; (vii) chuẩn hóa min-max và chuẩn hóa theo tỉ lệ Z; (viii) độ đo thập phân; (ix) chuyển dữ liệu về dạng chuẩn; (x) các phương pháp số cho phép xác định dữ liệu lẻ; (xi) chuyển các biến phạm trù sang biến số; (xii) gộp các biến số; (xiii) phân lớp các biến phạm trù; (xiv) loại bỏ biến không hữu ích; (xv) loại bỏ các bản ghi trùng. Liệt kê ra các nhiệm vụ này cho thấy có nhiều nhiệm vụ xử lý dữ liệu trong khoa học dữ liệu. Tuy không chi tiết vào các nhiệm vụ đó, cần biết sự cần thiết của chúng đối với quá trình khoa học dữ liệu.

### 2.1.1. Quá trình thăm dò dữ liệu

Trước hết là định nghĩa về thuật ngữ thăm dò dữ liệu.

*Định nghĩa: Thăm dò dữ liệu<sup>1</sup> là quá trình công tác thực hành nhiều bước, trong đó người dùng sử dụng phương pháp luận có cấu trúc để phát hiện và đánh giá các vấn đề phù hợp, xác định giải pháp và các chiến lược thực hiện, và sinh ra kết quả đo được.*

Mỗi bước có mục đích và chức năng riêng. Đối với quá trình, cần làm rõ (i) cách xác định mục tiêu tại mỗi bước; (ii) điều cần làm. Đối với một quá trình, cần xem những vấn đề liên quan đến giai đoạn trước, trong và sau quá trình. Có thể xem xét quá trình thăm dò dữ liệu theo (i) mức khái niệm, như đa số các nghiên cứu; (ii) kinh nghiệm thực hành, đề cập đến nhiều khía cạnh và mối quan hệ lẫn nhau giữa các bước.

Về đại thể, các bước trong quá trình thăm dò dữ liệu gồm:

Thăm dò không gian vấn đề;

1. Thăm dò không gian lời giải;

2. Xác định phương pháp thực hiện;

3. Khai phá dữ liệu. Bước này được chi tiết hóa theo ba phần (i) chuẩn bị dữ liệu; (ii) khảo sát dữ liệu; và (iii) mô hình hóa dữ liệu.

Ngoài việc có các tên xác định, các bước còn có các đặc trưng về (i) tỉ lệ thời gian thực hiện; (ii) mức quan trọng để quá trình thành công. Quá trình này được nhìn nhận như một dự án.

**Bảng 2.1. Các bước của dự án thăm dò dữ liệu và các đặc trưng**

Các bước	Thời gian hoàn thành (theo tỉ lệ so với tổng thời gian)	Mức quan trọng để dự án thành công
Tổng	20	80
Thăm dò không gian vấn đề;	10	15
Thăm dò không gian lời giải;	9	14

<sup>1</sup> Data exploration: thăm dò dữ liệu.

Xác định phương pháp thực hiện;	1	51
Tổng	80	20
<i>Khai phá dữ liệu:</i>		
Chuẩn bị dữ liệu;	60	15
Khảo sát dữ liệu;	15	3
Mô hình hóa dữ liệu.	5	2

Nhận xét về thông tin trong bảng, ba bước đầu sử dụng hết 20% thời gian tổng số, nhưng có mức quan trọng là 80%. Trong bước thứ tư, việc chuẩn bị dữ liệu chiếm nhiều thời gian và có mức quan trọng là 15%.

#### *2.1.1.1. Bước thăm dò không gian vấn đề*

Khi giải vấn đề cần xem xét môi trường quanh vấn đề đó. Chẳng hạn môi trường kinh doanh của doanh nghiệp, môi trường điều hành của một tổ chức... Dựa vào không gian đó mà người ta hiểu vấn đề cần giải và hiểu về dữ liệu cần thiết để giải vấn đề đó.

*Định nghĩa: Không gian vấn đề<sup>1</sup> là khung để xác định vấn đề và để phát hiện lời giải.*

#### *Xác định vấn đề*

Việc thăm dò dữ liệu bắt đầu với việc *xác định đúng vấn đề cần giải*. Chẳng hạn các chuyên gia truyền thông tiếp thu ý kiến khách hàng để cải thiện dịch vụ; các bác sĩ sử dụng thông tin từ bệnh nhân để xác định bệnh; người tạo nên tự điều chỉnh sản phẩm theo thị trường.

Mô hình dự báo có ích trong việc xác định vấn đề cần giải. Vấn đề được xác định đúng để tăng lợi nhuận từ phân khúc thị trường. Khi xây dựng được mô hình phù hợp, người ta có thể xác định đúng vấn đề, để tăng giá trị sản phẩm.

#### *Xác định vấn đề chính xác*

Liên quan đến thuật ngữ hiệu quả, để xác định vấn đề một cách hiệu quả, có các mức độ (i) xác định đúng vấn đề; (ii) xác định vấn đề chính xác. Một khi vấn đề được mô tả (i) cụ thể; (ii) rõ ràng; (iii) phù hợp với điều kiện về dữ liệu để giải, thì việc giải sẽ thuận lợi hơn.

*Định nghĩa: Giải vấn đề<sup>2</sup> là quá trình hay hành động phát hiện lời giải cho vấn đề.*

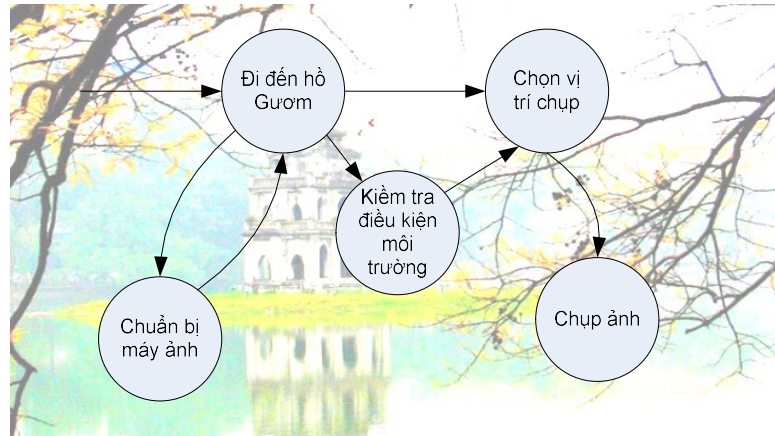
<sup>1</sup> Problem space: không gian vấn đề.

<sup>2</sup> Problem solving: giải vấn đề.

### *Bản đồ nhận thức*

Đối với tri thức, thể hiện, hay biểu diễn, tri thức là có ý nghĩa khi giải vấn đề. Bản đồ nhận thức là một tiếp cận biểu diễn tri thức về không gian vấn đề.

*Định nghĩa: Bản đồ nhận thức<sup>1</sup> là các hình ảnh liên quan đến trí não về các thuộc tính của môi trường xung quanh.*



**Hình 2.1. Bản đồ nhận thức về chụp ảnh hồ Gươm**

Các tiếp cận thể hiện trực quan khác là:

- Biểu đồ tuần tự, cho biết các sự kiện và mối liên giữa các sự kiện;
- Lược đồ nhân quả, tập trung vào các mối quan hệ giữa các số lượng, để biết sự kiện nào liên quan đến đích của vấn đề;
- Đối với vấn đề liên quan đến kinh doanh, người ta dùng sơ đồ khối, biểu đồ làn bơi, biểu đồ hệ thống.

### *Việc giải mơ hồ*

Trong quá trình giải vấn đề, trước tiên có thể đưa ra nhiều ứng viên cho lời giải, rồi lựa chọn lời giải mong muốn. Việc này thực hiện được do khớp các hình ảnh trong trí não về không gian vấn đề với các giả thiết liên quan. Như vậy không gian vấn đề sẽ rõ ràng hơn, tiện cho việc giải. Kỹ thuật này được gọi là giải mơ hồ.

Định nghĩa: Việc giải mơ hồ<sup>2</sup> là việc tìm kiếm lời giải trong điều kiện nói rộng, để xác định khoảng có lời giải, nhằm điều chỉnh điều kiện để tìm được lời giải cần thiết.

### *Xếp hạng theo cặp và xây dựng ma trận vấn đề*

Thăm dò không gian vấn đề tùy thuộc vào phạm vi dự án thăm dò, cho phép giải nhiều vấn đề. Lúc này nảy sinh câu hỏi về (i) loại vấn đề nào thích hợp đối với

<sup>1</sup> Cognitive map: bản đồ nhận thức.

<sup>2</sup> Ambiguity resolution: giải một cách mơ hồ.

quá trình thăm dò, tài nguyên sẵn có; (ii) loại vấn đề nào được giải với thời gian và tài nguyên hạn chế.

Theo lí thuyết quyết định và kinh tế lượng, có thể dùng xếp hạng cặp đôi để thấy sự tương ứng giữa lời giải và vấn đề. Xếp hạng này giúp cộng đồng quyết định thứ quan trọng nhất, nên bắt đầu công việc từ đâu.

*Định nghĩa: Xếp hạng theo cặp<sup>1</sup> là kĩ thuật xếp hạng ưu tiên hay liệt kê.*

Ma trận các vấn đề có dạng như khối thông tin, với hàng và cột, thường chứa các con số hay biểu thức đại số. Trong công nghệ thông tin người ta còn dùng thuật ngữ mảng hai chiều để chỉ ma trận.

Các vấn đề

Ô có giá trị 2 do có hai con số 3 cho vấn đề 3, tức đến xem. Giá trị này lớn nhất, nên cột xếp hạng có giá trị 1

Các vấn đề	1	2	3	điểm	hạng
1. Chụp ảnh	1	1	3	1	2
2. Vẽ tranh		1	3	0	3
3. Đến xem			3	2	1

Ô này có số 1, ứng với vấn đề 1, tức chụp ảnh, khi vấn đề 1 quan trọng hơn vấn đề 2, tức vẽ tranh

Ô này có số 3, ứng với vấn đề 3, tức đến xem, khi vấn đề 3 quan trọng hơn vấn đề 2, tức vẽ tranh

**Hình 2.2. Thí dụ về xếp hạng cặp đôi giữa ba vấn đề, khi cần biết một phong cảnh hồ Gươm**

#### 2.1.1.2. Bước thăm dò không gian lời giải

Việc tìm kiếm lời giải cho một vấn đề, tức giải vấn đề, được xác định như hoặc (i) chỉ ra thuật toán giải; hoặc (ii) cho thấy các bước giải; hoặc (iii) chỉ ra cách đi đến lời giải trong không gian các lời giải ứng với vấn đề đó.

*Định nghĩa: Không gian lời giải<sup>2</sup> của một vấn đề là tập các lời giải chấp nhận được cho vấn đề đó.*

Không gian lời giải cũng là thể hiện của tập các lời giải khả thi, hay tập các lời giải chấp nhận được đối với vấn đề tổ hợp, hay một loạt các khả năng giải vấn đề theo một cách riêng.

Đầu ra điển hình của dự án thăm dò dữ liệu đơn giản có dạng (i) báo cáo; (ii) biểu đồ; (iii) đồ thị; (iv) mã chương trình; (v) danh sách các bản ghi; (vi) công thức

<sup>1</sup> Pairwise ranking: xếp hạng theo cặp.

<sup>2</sup> Solution space: không gian lời giải.

đại số. Trong bước thăm dò không gian lời giải, lời giải thu được cần (i) chính xác; (ii) đủ phức tạp để phù hợp với thế giới thực. Liên quan đến tính đủ phức tạp: thể hiện lời giải theo cách đơn giản là tiêu chí ưu tiên; nhưng đối với thế giới thực, lời giải cần phủ các khía cạnh chính của thực tế, nên cần mức phức tạp vừa đủ. Đặc tả lời giải trong bước này cần thiết đối với quá trình thăm dò lời giải, chứ chưa là đặc tả đối với khai phá dữ liệu; nên đặc tả không quá phức tạp. Ngoài ra, đầu ra của quá trình này cần thực tế, tức việc thực hiện lời giải là khả thi.

Cả quá trình thăm dò vấn đề và quá trình thăm dò lời giải đều coi trọng việc giải mơ hồ. Kỹ thuật này dùng để thử nghiệm, nhằm tưởng tượng ra vấn đề nào và thực tế thu được vấn đề nào. Kỹ thuật cũng thử nghiệm để biết việc tưởng tượng ra lời giải và lời giải thực tế. Tuy nhiên trong thực tế, đối với nhiệm vụ thực, cần thiết khử tính mơ hồ trong các quá trình thăm dò không gian vấn đề và không gian lời giải.

#### *2.1.1.3. Bước xác định phương pháp thực hiện*

Đặc tả thực hiện là bước cuối cùng để chi tiết hóa các lời giải khác nhau cho vấn đề đã chọn, để người ta ứng dụng trong thực tế. Đặc tả hoàn toàn xác định lời giải, cho biết (i) vấn đề đề cập; (ii) những giá trị mà dự án thu được; (iii) người dùng sử dụng lời giải; (iv) hạn chế và kì vọng về lời giải; (v) thời gian đáp ứng yêu cầu về người dùng, cách thức, ra sao, khi nào, ở đâu<sup>1</sup>.

Xác định phương pháp thực hiện có mục đích (i) tăng lợi nhuận; (ii) tăng chất lượng; (iii) càng đáp ứng nhu cầu người dùng; (iv) giảm lãng phí; (v) giảm lỗi; (vi) đáp ứng các đích về kinh doanh đặc biệt. Dự án sẽ thất bại, mất lợi nhuận, nếu không được thực hiện nghiêm túc; tức bước này không được xác định đầy đủ.

#### *2.1.1.4. Bước khai phá dữ liệu*

Khai phá dữ liệu là quá trình nhiều bước. Trong bước chuẩn bị, đối tượng là dữ liệu; người khai phá làm việc với dữ liệu này. Nhiều dữ liệu trong bước chuẩn bị có thể được xử lí tự động; người khai phá tương tác với dữ liệu không xử lí tự động được.

Việc khảo sát là rất quan trọng để khai phá hiệu quả. Trong lúc khảo sát, người khai phá xác định sẽ dùng dữ liệu nào. Sau khi hoàn thành việc chuẩn bị và khảo sát, việc mô hình hóa dữ liệu trở nên đơn giản hơn. Các mô hình chỉ dùng để xử lí dữ liệu, chứ không làm dữ liệu sâu sắc và bộc lộ thêm. Các mô hình được xây dựng chỉ khi đã có kết luận về đầu ra.

---

<sup>1</sup> Who, how, what, when, where: ai, cách thức, ra sao, khi nào, ở đâu.



### Chuẩn bị dữ liệu cho mô hình hóa

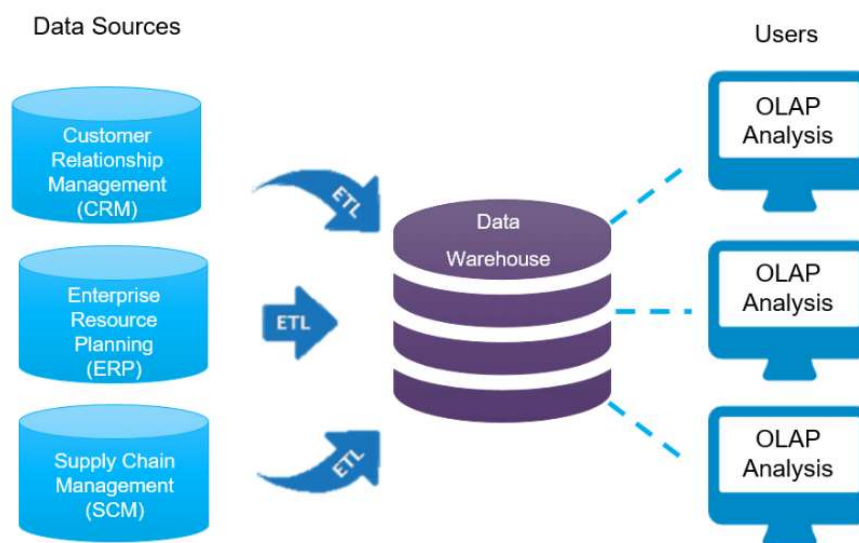
Chuẩn bị dữ liệu để người khai phá sẽ dùng nó trong việc chọn mô hình phù hợp. Có nhiều mô hình khai phá dữ liệu như mô hình suy luận, mô hình dự báo... sử dụng nhiều kĩ thuật, nhưng vẫn lấy dữ liệu làm trọng, quán triệt nguyên tắc GIGO<sup>1</sup>. Việc dự báo<sup>2</sup> là nhiệm vụ khai phá dữ liệu.

*Định nghĩa: Người khai phá<sup>3</sup> là người phát hiện các mẫu trong tập dữ liệu<sup>4</sup> lớn, nhờ phương pháp kết hợp giữa học máy, thống kê, cơ sở dữ liệu. Họ thực hiện quá trình trí tuệ để trích ra các mẫu.*

Người ta nhận thấy có nhiều phương pháp luận và minh chứng thực tế đối với việc dựng mô hình với nhiều thuật toán đa dạng; nhưng không có các phương pháp luận tương tự đối với việc chuẩn bị dữ liệu. Bởi vậy, việc chuẩn bị dữ liệu càng quan trọng để mô hình hóa thực sự trong thế giới thực.

Việc chuẩn bị dữ liệu cần được thực hiện đến mức phục vụ tối đa cho mô hình hóa. Dữ liệu cần giữ được bản chất tự nhiên, nhất là tương quan giữa các phần tử dữ liệu, để phù hợp với người khai phá. Việc này khác với việc thu thập dữ liệu cho kho dữ liệu.

*Định nghĩa: Kho dữ liệu<sup>5</sup> là hệ thống cho phép lập báo cáo và phân tích dữ liệu, như hệ thống thông minh doanh nghiệp BI<sup>6</sup>.*



**Hình. Kho dữ liệu**

<sup>1</sup> GIGO: Garbage In, Garbage Out: vào rác, ra rác.

<sup>2</sup> Prediction: dự báo, dự đoán; đề nghị sử dụng thuật ngữ dự báo.

<sup>3</sup> Miner: người khai phá.

<sup>4</sup> Data set: đề nghị sử dụng thuật ngữ tập dữ liệu.

<sup>5</sup> Datawarehouse: kho dữ liệu.

<sup>6</sup> BI: Business Intelligence: thông minh doanh nghiệp, là hệ thống được phát triển như hệ thống trợ giúp quyết định DSS.

### *Chuẩn bị môi trường thông tin*

Một mục đích khác của việc chuẩn bị dữ liệu là xác định môi trường thông tin đã chuẩn bị. Nhiệm vụ chính của môi trường này là bảo vệ công cụ mô hình hóa khỏi các dữ liệu có lỗi và thể hiện tối đa nội dung thông tin của dùng cho công cụ mô hình hóa. bộ dữ liệu

*Định nghĩa: Môi trường thông tin đã chuẩn bị PIE<sup>1</sup> là chương trình tính toán động, bao các công cụ mô hình hóa, tránh các mô hình khỏi các dữ liệu biến dạng, sai sót.*

Cần phân biệt giữa các tập dữ liệu dùng trong khai phá dữ liệu. Các tập dữ liệu huấn luyện, thử nghiệm hay khai thác đều rút ra từ dữ liệu dùng để mô hình hóa. Đôi khi tập dữ liệu khai thác không rút ra vào lúc mô hình hóa.

*Định nghĩa: Tập dữ liệu huấn luyện<sup>2</sup> là các trường hợp để học, khớp với các tham số, tức trọng số, của bộ phân lớp. Tập dữ liệu thử nghiệm<sup>3</sup> là tập dữ liệu độc lập với tập dữ liệu huấn luyện, nhưng có cùng phân phối xác suất với tập dữ liệu huấn luyện. Tập dữ liệu xác nhận<sup>4</sup> là tập các trường hợp dùng để tinh chỉnh các siêu tham số, tức tham số về kiến trúc, của bộ phân loại.*

Môi trường PIE có các thành phần:

- *Khối nhập PIE-I* có vai trò vùng đệm thông minh giữa (i) dữ liệu nhập vào; (ii) tập dữ liệu huấn luyện, thử nghiệm, khai thác. Do các biến dự báo được PIE-I chuẩn bị dữ liệu, nên bất kì dự báo mô hình nào đều sử dụng dữ liệu được chuẩn bị. Các dự báo được chuyển sang dạng phù hợp với khối xuất PIE-O;
- *Khối xuất PIE-O*. Các biến dự báo của mô hình và những thứ mà mô hình cổ dự báo hay giải thích sẽ được chuyển sang dạng phù hợp với tập dữ liệu huấn luyện.

Nói chung, việc chuẩn bị dữ liệu để mô hình hóa đòi hỏi nhiều điều chỉnh để dữ liệu tốt cho việc mô hình hóa. Mô hình thu được dựa trên dữ liệu đã điều chỉnh. Cần có cơ chế đảm bảo bất kì dữ liệu mới, đặc biệt dữ liệu mà mô hình sẽ áp lên, sẽ được điều chỉnh tương tự như dữ liệu huấn luyện, tức có phân phối xác suất tương tự. Nếu không đảm bảo được điều này thì mô hình sẽ không có giá trị như với dữ liệu thô. Vai trò chính của PIE là chuyển hóa dữ liệu để các dữ liệu mới sẽ có phân phối như dữ liệu huấn luyện, tức chuyển dữ liệu mới sang dạng đã biết khi mô hình hóa.

---

<sup>1</sup> PIE: Prepared Information Environment: môi trường thông tin đã chuẩn bị.

<sup>2</sup> Training dataset: tập dữ liệu huấn luyện.

<sup>3</sup> Test dataset: tập dữ liệu thử nghiệm.

<sup>4</sup> Phân biệt *validation* (xác nhận) với *verification* (xác minh). *Xác nhận* là việc đánh giá theo sản phẩm cuối cùng, đối chiếu với sản phẩm mẫu; *xác minh* là việc đánh giá sản phẩm, quá trình, để biết tính đúng đắn của quá trình.

Đối với mô hình hóa một lần, mà các dữ liệu được mô hình hóa và được giải thích đều đã biết, vai trò của môi trường PIE là hạn chế. PIE chỉ đơn giản sinh ra tập dữ liệu để mô hình hóa. Do đã biết tất cả dữ liệu, PIE chỉ chuyển các biến đầu ra từ giá trị đã điều chỉnh dự báo được sang giá trị dự báo mong đợi thực.

Đầu ra mong đợi từ quá trình chuẩn bị dữ liệu là (i) người khai phá; (ii) tập dữ liệu đã chuẩn bị; (iii) môi trường PIE, cho phép mô hình đã huấn luyện áp lên các tập dữ liệu khác và thực hiện nhiều chức năng khác. Môi trường PIE đảm bảo chứa quanh mô hình này, cả khi huấn luyện và khai thác, để tách mô hình này khỏi vấn đề với dữ liệu thô đã dùng khi chuẩn bị dữ liệu.

### *Khảo sát dữ liệu*

*Định nghĩa: Khảo sát<sup>1</sup> là mô tả về chủ đề, ghi lại chi tiết về chủ đề đó, xem xét cấu trúc của chủ đề.*

Việc khảo sát tập trung trả lời ba câu hỏi (i) có gì trong tập dữ liệu; (ii) cần thu được câu trả lời cho câu hỏi; (iii) miền nào là nguy hiểm. Các câu hỏi này cũng tương tự như các câu hỏi đặt ra cho quá trình mô hình hóa, nhưng mang ý nghĩa khác.

Việc khảo sát nhìn tập dữ liệu khác về bản chất so với tiếp cận mô hình hóa dữ liệu. (i) Việc mô hình hóa tối ưu câu trả lời từ vấn đề cụ thể, chuyên sâu nào đó. Phát hiện một hay nhiều vấn đề phù hợp nhất thuộc về bước đầu tiên của quá trình thăm dò dữ liệu. Đưa ra các câu trả lời là vai trò của bước mô hình hóa của quá trình khai phá dữ liệu; (ii) Việc khảo sát nhìn cấu trúc tổng quát của dữ liệu và báo cáo về hiện diện của thông tin hữu dụng trong tập dữ liệu. Việc khảo sát không thực sự liên quan tới đúng thông tin đảm bảo cho mô hình hóa. Mục đích cụ thể nhất của khảo sát là phát hiện xem câu trả lời cho vấn đề đã mô hình hóa có là dữ liệu ưu tiên đầu tư cho việc xây dựng mô hình này hay không.

Với tập dữ liệu phong phú, việc khảo sát sẽ nhìn sâu vào các mối quan hệ và các mẫu tổng quát trong dữ liệu. Nó không cố giải thích hay đánh giá dữ liệu, nhưng chỉ ra cấu trúc dữ liệu. Việc mô hình hóa thăm dò cấu trúc mịn; còn việc khảo sát vạch ra cấu trúc rộng hơn.

Xác định miền nguy hiểm đối với mô hình là cần thiết, để tại đó mô hình sẽ hiệu quả.

*Định nghĩa: Miền nguy hiểm<sup>2</sup> là miền dữ liệu có mối quan hệ thay đổi nhanh, và dữ liệu không mô tả tốt, gây nên nghi ngờ đối với hiệu quả của mô hình.*

---

<sup>1</sup> Survey: khảo sát.

<sup>2</sup> Danger area: miền nguy hiểm.

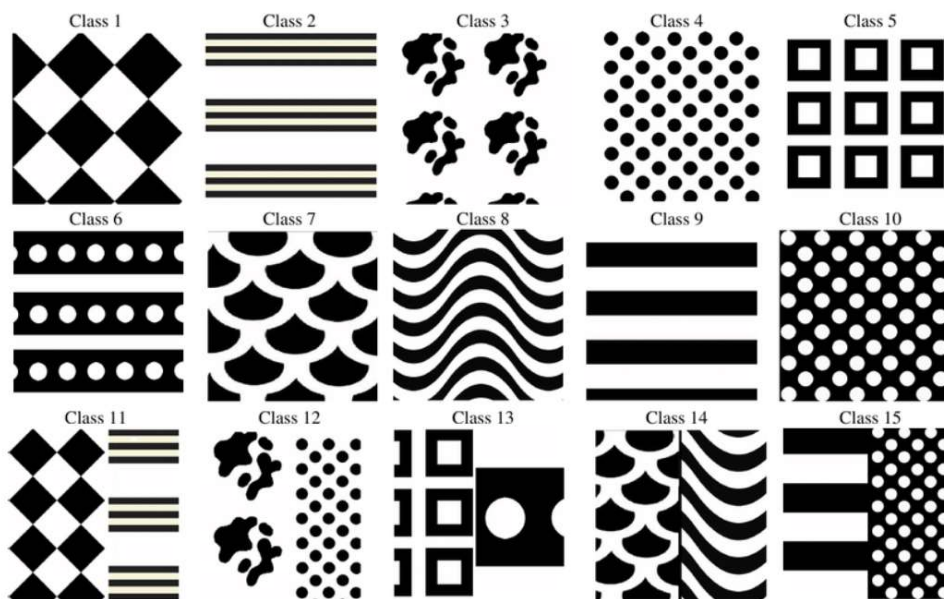
Việc khảo sát vạch ra bản đồ khái quát cho việc thăm dò dữ liệu. Trong quá trình thăm dò dữ liệu, bản đồ này còn được tinh chỉnh. Tuy vậy, việc có bản đồ định hướng là quan trọng.

### *Mô hình hóa dữ liệu*

Trong khai phá dữ liệu, cần phân biệt mẫu với mô hình.

*Định nghĩa: Mẫu<sup>1</sup> là qui định, hình dạng thấy ngay được trong thế giới hay trong thiết kế của con người, lặp lại theo cách định trước. Mô hình<sup>2</sup> là khuôn mẫu áp dụng cho toàn thể các đối tượng, là thể hiện của hệ thống.*

Nếu tập trung vào các công cụ mô hình hóa việc khai phá dữ liệu ngay từ đầu, khi tiếp cận vấn đề, người ta thường làm biến dạng vấn đề theo dạng không phù hợp. Công cụ khai phá không phải là ưu tiên đầu tiên, mà phải là mô hình để khai phá. Do vậy, người ta không đề cập đến công cụ cụ thể, chẳng hạn vấn đề nào để sử dụng mạng nơ ron, mà đề cập đến cách làm và lí do thực hiện trên mạng nơ ron.



**Hình. Mẫu dùng trong mô hình**

#### *2.1.1.5. Thăm dò tức khai phá và mô hình hóa*

Quá trình thăm dò dữ liệu có nhiều bước; không bước nào độc lập. Người ta xác định vấn đề, rồi xác định các lời giải tiềm năng, rồi phát hiện và chuẩn bị dữ liệu phù hợp, qua việc khảo sát và mô hình hóa. Các bước đều liên hệ với nhau.

Việc mô hình hóa, các dạng công cụ và dạng mô hình có quan hệ chặt chẽ với cách chuẩn bị dữ liệu.

<sup>1</sup> Pattern: mẫu. Cần phân biệt với sample.

<sup>2</sup> Model: mô hình.

### 2.1.2. Khai phá dữ liệu, mô hình hóa và các công cụ mô hình hóa

Mục đích chính của chuẩn bị dữ liệu là để thăm dò ra mô hình. Dữ liệu chuẩn bị được sẽ mang thông tin; thông tin có ích nếu người ta có thể hiểu được những thứ mà thông tin đem lại. Do vậy việc mô hình hóa cần chuyển những thông tin trong dữ liệu thành dạng mà con người nhận thức được.

#### 2.1.2.1. Mười qui tắc vàng

Việc chuẩn bị dữ liệu giúp xây dựng một khung cho việc khai phá dữ liệu, để các công cụ thích hợp sẽ dùng cho dữ liệu thích hợp; các dữ liệu này đã được chuẩn bị theo vấn đề kinh doanh và cho ra lời giải cần thiết. Khung này cần thiết đối với người khai phá, để cho ra kết quả tốt nhất cho dự án khai phá dữ liệu. Đối với việc xây dựng các mô hình, có mười qui tắc vàng đi cùng với khung này.

1. Chọn các vấn đề được xác định rõ ràng, để đạt được lợi nhuận hữu hình;
2. Xác định lời giải đã yêu cầu;
3. Xác định cách sử dụng lời giải thu được;
4. Hiểu tối đa về vấn đề và tập dữ liệu;
5. Để vấn đề điều khiển việc mô hình hóa, tức việc chọn công cụ, chuẩn bị dữ liệu;
6. Thiết lập giả thiết;
7. Tinh chỉnh mô hình nhiều lần;
8. Làm đơn giản mô hình. Qui tắc được xem như KISS<sup>1</sup>;
9. Xác định điều bất ổn trong mô hình, tức xác định miền nguy hiểm;
10. Xác định điều không chắc chắn trong mô hình, tức xác định miền nguy hiểm và dải dữ liệu mà mô hình cho ra dự báo ít tin cậy.

Xây dựng một mô hình dữ liệu là diễn tả các mối quan hệ thay đổi trong một biến, hay tập các biến, theo một biến khác, hay tập các biến khác. Hoặc không để ý đến loại mô hình, đích của mô hình là diễn tả, theo thuật ngữ kí hiệu, hình dạng về cách thay đổi của một biến, hay tập các biến, khi một biến hay tập các biến khác thay đổi; việc này giúp thu được thông tin về độ tin cậy của mối quan hệ giữa các biến. Có nhiều cách thể hiện mối quan hệ; thông dụng nhất là thể hiện mối quan hệ theo (i) đồ thị; (ii) phương trình toán học; (iii) chương trình máy tính.

*Định nghĩa: Mô hình bị động<sup>2</sup> là mô hình thể hiện các mối quan hệ hay các liên kết có trong tập dữ liệu. Mô hình chủ động<sup>3</sup> là mô hình nhận các đầu vào mẫu và cho ra dự báo về đầu ra mong muốn.*

---

<sup>1</sup> KISS: Keep It Sufficiently Simple, giữ đủ đơn giản.

<sup>2</sup> Passive model: mô hình bị động.

<sup>3</sup> Active model: mô hình chủ động.

Cho dù các mô hình được xây dựng để đáp ứng nhiều yêu cầu, nhưng mục tiêu trong khai phá dữ liệu là cho ra hoặc (i) mô hình dự báo; hoặc (ii) mô hình giải thích, tức mô hình suy luận.

#### 2.1.2.2. Giới thiệu các công cụ mô hình hóa

Các công cụ mô hình hóa là đa dạng. Các công cụ còn phân tích để đưa ra nhiều mô hình ở các dạng khác nhau; là phần mềm máy tính có chức năng (i) chuyển đổi; (ii) xử lý tập dữ liệu. Có công cụ thực hiện tự động, không cần đến tri thức chuyên gia, tri thức lĩnh vực cần thiết. Cũng có công cụ thực hiện như hệ thống trợ giúp, với cơ sở dữ liệu và kho dữ liệu.

Do khai phá dữ liệu là thăm dò các mẫu dùng cho tình huống kinh doanh, nên có công cụ phân tích thống kê có ý nghĩa. Ranh giới giữa phân tích thống kê và khai phá dữ liệu là khó phân biệt, ngoài quan điểm triết học về hai nhiệm vụ này. Chính vì thế mà không cần tách biệt công cụ dùng cho lĩnh vực nào.

- Theo quan điểm triết học và lịch sử, *phân tích thống kê* hướng đến việc xác minh và xác nhận giả thuyết. Người ta đưa ra giả thuyết và tìm các sự kiện để xem có thể chấp thuận hay từ bỏ giả thuyết này. Lập luận<sup>1</sup> thống kê liên quan đến chứng minh logic; và như hệ thống hình thức, nó không liên quan đến tầm quan trọng hay tác động của kết quả. Điều này có nghĩa trong trường hợp cực đoan, nó đưa ra kết quả có giá trị thống kê, nhưng không có nghĩa.
- Với *khai phá dữ liệu*, thay vì người thực nghiệm sắp sẵn giả thuyết và kiểm nghiệm các sự kiện, người ta thực hiện xoay, lật vấn đề. Với các tham số của quá trình thăm dò dữ liệu, việc khai phá dữ liệu tiếp cận đến tập các dữ liệu và tìm đâu là giả thuyết mà dữ liệu trợ giúp được. Có sự khác nhau lớn về khái niệm: nhiều giả thiết được đặt ra để khai phá dữ liệu là không có ý nghĩa, không liên kết với tính sử dụng hay có giá trị. Điều này có nghĩa rằng với khai phá dữ liệu, người dùng có (i) tập các ý tưởng khá đầy đủ; (ii) các liên kết; (iii) các ảnh hưởng... Công việc khi đó là cảm nhận dữ liệu và sử dụng. Với phân tích thống kê thì trước hết người dùng lập ra ý tưởng, liên kết, ảnh hưởng để thử nghiệm.
- Có lĩnh vực thống kê, gọi là *phân tích dữ liệu thăm dò*<sup>2</sup>, sử dụng dấu hiệu phân định ranh giới giữa phân tích thống kê và khai phá dữ liệu. Phân tích thống kê sử dụng nhiều công cụ cho phép trí tuệ con người hiển thị và lượng hóa các mối quan hệ vốn có trong dữ liệu, để có khả năng tìm kiếm mẫu. Trước đây người ta hay thực hiện điều này. Ngày nay, lượng

<sup>1</sup> Phân biệt *reasoning*: lập luận; *deduction*: suy diễn; *inference*: suy luận.

<sup>2</sup> Exploration data analysis: phân tích dữ liệu thăm dò.



dữ liệu nhiều, cho phép định lượng với khả năng hơn con người, nên người ta sử dụng giải pháp tự động hóa, với kĩ thuật *học máy*. Học máy cho phép học được các mẫu có sẵn trong tập dữ liệu. Việc kiểm định ý nghĩa của đầu ra vẫn do con người thực hiện, tức con người vẫn điều khiển chương trình để huấn luyện và học các mẫu.

*Định nghĩa: Học máy<sup>1</sup> là một lĩnh vực trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống học tự động từ dữ liệu để giải quyết những vấn đề cụ thể.*

#### *2.1.2.3. Các kiểu mô hình*

Liên quan đến xây dựng mô hình và cho ra kết quả, có một số người ít kinh nghiệm đã xem việc mô hình hóa là quá trình tuần tự. Đó là quá trình (i) thiết lập vấn đề; (ii) chọn công cụ; (iii) lấy dữ liệu; dựng mô hình; (iv) ứng dụng mô hình; (v) đánh giá kết quả. Ngược lại, *việc xây dựng mô hình là quá trình liên tục, tích hợp vài lần quay vòng và có tương tác ở mỗi thành phần*. Tại mỗi bước, có kiểm tra để đảm bảo mô hình thực sự đáp ứng các mục tiêu. Đó là quá trình động, lặp để đến lời giải tốt, cần đến tương tác để người dùng hướng dẫn đến lời giải tối ưu.

#### *2.1.2.4. Các mô hình tĩnh và mô hình động*

Về cơ bản, mô hình chủ động thực sự có tương tác, theo cách nào đó; còn mô hình thụ động không tương tác.

*Định nghĩa: Mô hình chủ động là mô hình có tương tác, cần dữ liệu đầu vào. Mô hình bị động là mô hình không nhận dữ liệu đầu vào.*

- Các mô hình thụ động trả lời các câu hỏi và cho biết các mối quan hệ nhờ biểu đồ, đồ thị, văn bản, công thức toán học... Mô hình diễn tả theo cách nó hiểu về nguyên nhân của các mối quan hệ. Mô hình là bị động ở chỗ không lấy đầu vào, mà đưa ra đầu ra, thay đổi, phản ứng khi nó hoạt động;
- Mô hình chủ động tương tác với người dùng, thu thập thêm dữ liệu, có một hay nhiều hoạt động.

Sự khác nhau giữa các mô hình chủ động và bị động là không tiện cho người mô hình hóa và ứng dụng. Cũng có tác động đáng kể trong việc chọn dữ liệu để mô hình hóa. Tuy nhiên khi chuẩn bị tập dữ liệu, cần đề cập đến sự khác nhau giữa mô hình chủ động với bị động nếu kiểu mô hình có tác động đến việc chuẩn bị dữ liệu để mô hình hóa.

---

<sup>1</sup> Machine learning: học máy.

#### *2.1.2.5. Các mô hình dự báo và mô hình giải thích*

Có mô hình cho phép giải thích hiện tượng dựa trên các dữ liệu cụ thể; cũng có mô hình dự báo, phân lớp hay giải thích dữ liệu. Phân biệt này khác với phân biệt về chủ động hay bị động.

Đôi khi có mô hình vừa được xem là mô hình dự báo, mô hình bị động, như trong tự động hóa công nghiệp. Trong kinh doanh có dạng mô hình quan sát hành động của chuyên gia để mô hình hóa, thay vì yêu cầu chuyên gia tạo mô hình. Tiếp cận này gọi là *lấy tri thức chuyên gia về tự động*<sup>1</sup>. Cũng có mô hình vừa chủ động, vừa giải thích, hay mô hình vừa bị động, vừa giải thích...

*Định nghĩa: Mô hình dự báo*<sup>2</sup> là mô hình sử dụng thống kê để dự báo đầu ra. *Mô hình giải thích*<sup>3</sup> là mô hình mô tả lý do cách thức làm việc của đối tượng hay giải thích lý do của hiện tượng.

#### *2.1.2.6. Các mô hình tĩnh và mô hình học liên tục*

##### *Các mô hình tĩnh*

*Định nghĩa: Mô hình tĩnh*<sup>4</sup> là mô hình phát hiện các mối quan hệ hay trả lời các câu hỏi từ dữ liệu lịch sử.

Thuật ngữ dữ liệu lịch sử có nghĩa tập dữ liệu dùng để xây dựng mô hình không là dữ liệu đã cập nhật như dữ liệu hiện tại. Trong trường hợp phát sinh lỗi của mô hình, do không đáp ứng được thay đổi của dữ liệu, cần thiết thu thập thêm dữ liệu mới để xây dựng lại mô hình.

Khi trả lời cho các câu hỏi người dùng, các mô hình tĩnh cũng khá phức tạp do làm việc với nhiều tài nguyên, các công cụ, và do năng lực của người xây dựng mô hình. Người mô hình hóa ít kinh nghiệm thường thấy mô hình tĩnh như cách diễn ra quá trình mô hình hóa, tức người ta mô hình hóa theo thói quen, mà không để ý đến các kỹ thuật phù hợp hơn.

##### *Các mô hình học liên tục*

Mô hình học liên tục thể hiện quá trình phát hiện, điều khiển, tương đối quản lý được trong điều kiện động. Việc kiến thiết một mô hình học liên tục bền vững cần đến các tài nguyên từ bên ngoài lĩnh vực thăm dò dữ liệu. Hạt nhân và công nghệ của việc học liên tục là khai phá dữ liệu với định hướng của quá trình thăm dò dữ liệu.

---

<sup>1</sup> Automated expertise capture: lấy tri thức chuyên gia về tự động.

<sup>2</sup> Prediction model: mô hình dự báo.

<sup>3</sup> Explanatory model: mô hình giải thích.

<sup>4</sup> Static model: mô hình tĩnh.



*Định nghĩa: Học liên tục<sup>1</sup> là việc học sử dụng mô hình tự trị, tự nói rộng kĩ năng thông qua việc học và tri thức gia tăng.*

Việc học liên tục được ví như quá trình cuộc sống tự điều chỉnh nhu cầu để thích nghi cả về các nhân lẫn chuyên môn, khi thực tế thay đổi. Hệ thống học sử dụng mô hình tự trị, gồm nhiều *điểm đặt*<sup>2</sup> bên trong.

Trong hệ thống học liên tục nhân tạo, tập các điểm đặt ban đầu luôn được các yếu tố bên ngoài hệ thống xác định; các hệ thống học liên tục tự nhiên có thể tiến hóa để tự có các điểm đặt. Hệ thống đánh giá dữ liệu nhập vào và thay đổi hành vi hệ thống theo cách thay đổi các tham số về môi trường hệ thống; các tham số này ít nhiều chịu sự điều khiển để hệ thống nắm được các điểm đặt. Đó là việc điều chỉnh hệ thống tự thích nghi theo thời gian thực trước môi trường động. Và đó cũng là việc thay đổi liên tục cấu trúc bên trong để phản ánh kinh nghiệm quá khứ và sử dụng kinh nghiệm quá khứ để thay đổi môi trường của nó. Việc này khác với một loạt thao tác cập nhật của các mô hình tĩnh; vấn đề là trong mô hình học liên tục có sự tương tác liên tục giữa các thành phần của hệ thống.

Khi sử dụng hệ thống học liên tục, môi trường PIE cho phép việc chuẩn bị dữ liệu tự động, liên tục trong toàn bộ quá trình.

Về quá trình thăm dò dữ liệu, có nhận xét sau:

- Liên quan đến thăm dò dữ liệu không nên chỉ nghĩ đến công dụng của các công cụ đa dạng. Cần đề cập trước tiên đến chuẩn bị dữ liệu, để thăm dò dữ liệu có hiệu quả. Về chuẩn bị dữ liệu, người ta thấy thăm dò dữ liệu cần phù hợp với tập dữ liệu, với mô hình. Việc mô hình hóa liên quan nhiều đến quá trình chuẩn bị dữ liệu;
- Nét chính của thăm dò dữ liệu không tách biệt khỏi vấn đề kinh doanh mà người ta cần tìm lời giải. Thăm dò dữ liệu phục vụ cho nhu cầu của kinh doanh. Liên quan đến vấn đề kinh doanh là xác định vấn đề thực tế, rồi mới xác định dữ liệu phù hợp.

## 2.2. Quá trình khoa học dữ liệu

Sau đây trình bày:

- Hiểu các luồng của quá trình khoa học dữ liệu;
- Thảo luận về các bước trong quá trình khoa học dữ liệu.

Mục tiêu của phần này là giới thiệu tổng quan về *quá trình*<sup>3</sup> khoa học dữ liệu, nhưng chưa đi sâu vào dữ liệu lớn. Nội dung liên quan đến (i) tập dữ liệu lớn; (ii)

---

<sup>1</sup> Continuous learning: học liên tục.

<sup>2</sup> Set point: điểm đặt.

<sup>3</sup> Process: quá trình, với nghĩa qui định; quá trình, với nghĩa diễn biến theo thời gian.

dữ liệu truyền trực tuyến; (iii) dữ liệu văn bản sản xuất được đề cập trong các chương khác.

### 2.2.1 Tổng quan về quá trình khoa học dữ liệu

Tuân theo phương pháp tiếp cận có cấu trúc đối với khoa học dữ liệu sẽ giúp tối đa hóa cơ hội thành công trong một dự án khoa học dữ liệu với chi phí thấp nhất. Nó cũng giúp thực hiện một dự án như một nhóm, với mỗi thành viên trong nhóm tập trung vào những gì họ làm tốt nhất. Tuy nhiên, mỗi cách tiếp cận có thể không phù hợp với mọi loại dự án, không là giải pháp duy nhất để thực hiện tốt khoa học dữ liệu.

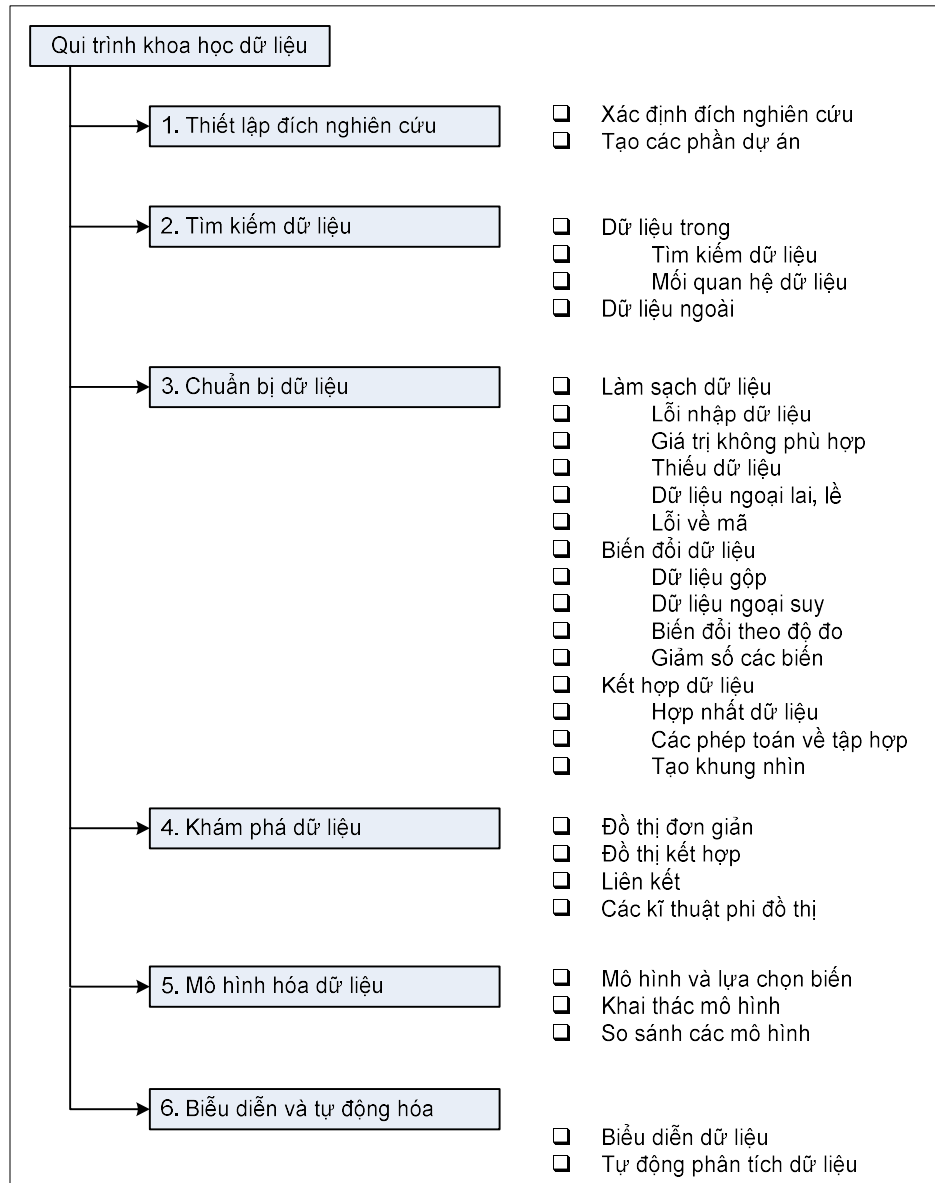
Quá trình khoa học dữ liệu điển hình bao gồm sáu bước, có lặp lại, như thể hiện trong hình 2.3. Nó cho thấy quá trình khoa học dữ liệu và hiển thị các bước và hành động chính mà sẽ thực hiện trong một dự án. Danh sách sau đây là một giới thiệu ngắn; mỗi bước sẽ được thảo luận sâu hơn.

1. Bước đầu tiên của quá trình này là *đặt mục tiêu nghiên cứu*. Mục đích chính ở đây là đảm bảo tất cả các bên liên quan hiểu được những điều liên quan *lí do, thế nào, ra sao*<sup>1</sup>. Trong mọi dự án nghiêm túc, điều này sẽ dẫn đến qui định của dự án;
2. Giai đoạn thứ hai là *tìm kiếm dữ liệu*. Cần có sẵn dữ liệu để phân tích, vì vậy bước này bao gồm việc tìm kiếm dữ liệu *phù hợp* và nhận quyền truy cập vào dữ liệu từ chủ sở hữu dữ liệu. Kết quả là dữ liệu ở dạng thô, có thể cần được *làm tinh* và *chuyển đổi* trước khi đưa vào sử dụng;
3. Tiếp theo là *chuẩn bị* dữ liệu. Việc này chuyển đổi dữ liệu từ dạng thô thành dữ liệu có thể sử dụng trực tiếp trong các mô hình. Để đạt được điều này, cần (i) phát hiện và sửa các loại lỗi khác nhau trong dữ liệu; (ii) kết hợp dữ liệu từ các nguồn dữ liệu khác nhau; (iii) biến đổi dữ liệu;
4. Bước thứ tư là *khám phá dữ liệu*. Mục tiêu của bước này là hiểu sâu về dữ liệu. Người ta sẽ tìm kiếm các (i) *mẫu*<sup>2</sup>; (ii) mối tương quan; (iii) độ lệch dựa trên các kĩ thuật mô tả và hình ảnh;

---

<sup>1</sup> Why, what, how: tại sao, thế nào, ra sao.

<sup>2</sup> Pattern: mẫu. Mẫu có phạm vi hẹp hơn mô hình.



**Hình 2.3. Quá trình khoa học dữ liệu**

- Nội dung về xây dựng mô hình, hay *mô hình hóa dữ liệu*, là quan trọng. Người ta cố gắng hiểu biết sâu sắc hoặc đưa ra các dự đoán được nêu trong quy định dự án. Sự kết hợp của các mô hình đơn giản có xu hướng tốt hơn một mô hình phức tạp;
- Bước cuối cùng của mô hình khoa học dữ liệu là *trình bày kết quả và tự động hóa phân tích*. Một mục tiêu của dự án là thay đổi quá trình và/hoặc đưa ra quyết định tốt hơn. Vấn đề thuyết phục doanh nghiệp rằng những phát hiện trong dự án sẽ thực sự thay đổi quá trình kinh doanh như mong đợi. Tầm quan trọng của bước này rõ ràng hơn trong các dự án ở cấp độ chiến lược và chiến thuật. Các dự án nhất định yêu cầu thực

hiện quá trình kinh doanh lặp đi lặp lại, vì vậy việc tự động hóa dự án sẽ tiết kiệm thời gian.

Trên thực tế, người ta sẽ không tiến triển một cách tuyến tính từ bước 1 đến bước 6. Thường thì sẽ đệ qui và lặp lại giữa các giai đoạn khác nhau.

Thực hiện theo sáu bước này có lợi về tỉ lệ thành công của dự án cao hơn và tăng tác động của kết quả nghiên cứu. Quá trình này đảm bảo một kế hoạch nghiên cứu được xác định rõ ràng, hiểu rõ về câu hỏi kinh doanh và phân phối rõ ràng trước khi bắt đầu xem xét dữ liệu. Các bước đầu tiên của quá trình tập trung vào việc lấy dữ liệu chất lượng cao làm đầu vào cho các mô hình. Bằng cách này, các mô hình sẽ hoạt động tốt hơn sau này. Trong khoa học dữ liệu, có một câu nói: *vào rác, ra rác*. Một lợi ích khác của việc làm theo cách tiếp cận có cấu trúc là làm việc nhiều hơn với dự án *mẫu*<sup>1</sup> trong khi tìm kiếm mô hình tốt nhất. Khi xây dựng một *mẫu*, có thể sẽ thử nhiều mô hình và sẽ không tập trung nhiều vào các vấn đề như tốc độ chương trình hoặc viết mã theo chuẩn. Thay vào đó, điều này cho phép người ta tập trung vào việc mang lại giá trị kinh doanh.

Không phải dự án nào cũng do doanh nghiệp tự khởi xướng. Những hiểu biết sâu sắc học được trong quá trình phân tích hoặc sự xuất hiện của dữ liệu mới có thể tạo ra các dự án mới. Khi nhóm khoa học dữ liệu tạo ra một ý tưởng, công việc đã được thực hiện để đưa ra đề xuất và tìm nhà tài trợ kinh doanh.

Chia dự án thành các giai đoạn nhỏ hơn cũng cho phép nhân viên làm việc cùng nhau như một nhóm. Người ta không thể là chuyên gia trong mọi lĩnh vực. Cần biết cách tải tất cả dữ liệu lên tất cả các cơ sở dữ liệu khác nhau, tìm một lược đồ dữ liệu tối ưu không chỉ hoạt động cho ứng dụng của mình mà còn cho các dự án khác trong công ty, sau đó (i) theo dõi tất cả các kĩ thuật thống kê và khai phá dữ liệu; đồng thời quan tâm đến (ii) các công cụ thuyết trình; (iii) chính sách kinh doanh. Đó là một nhiệm vụ khó khăn và đó là lí do ngày càng nhiều công ty dựa vào một nhóm chuyên gia hơn là cố gắng tìm một người có thể làm tất cả.

Quá trình mô tả trong phần này là phù hợp nhất đối với một dự án khoa học dữ liệu, chỉ chứa một vài mô hình. Nó không phù hợp với mọi loại dự án. Tuy nhiên, một nhà khoa học dữ liệu mới bắt đầu sẽ phải trải qua một chặng đường dài sau cách làm việc này.

#### *2.2.1.1. Không phụ thuộc vào quá trình*

Không phải mọi dự án sẽ tuân theo kế hoạch chi tiết này, bởi vì quá trình tùy thuộc vào sở thích của (i) người làm khoa học dữ liệu; (ii) công ty; (iii) bản chất của dự án. Một số công ty có thể yêu cầu đối tác công nghệ thông tin tuân theo một quá

---

<sup>1</sup> Prototype: nguyên mẫu, mẫu.

trình nghiêm ngặt, trong khi những công ty khác có cách làm việc thân mật hơn. Nói chung, sẽ cần một cách *tiếp cận có cấu trúc* khi làm việc trong một dự án phức tạp hoặc khi có nhiều người hoặc nhiều nguồn lực tham gia.

Mô hình dự án *nhANH, linh hoạt* là một thay thế cho một quá trình tuần tự với các lần lặp lại. Vì phương pháp này giành được nhiều chỗ đứng hơn trong công nghệ thông tin và trong toàn công ty, nên nó cũng đang được cộng đồng khoa học dữ liệu áp dụng. Mặc dù phương pháp nhanh phù hợp với dự án khoa học dữ liệu, nhưng nhiều chính sách của công ty sẽ ưu tiên cách tiếp cận cứng nhắc hơn đối với khoa học dữ liệu. Không phải lúc nào cũng có thể lập kế hoạch về mọi chi tiết của quá trình khoa học dữ liệu và thường xuyên hơn là lặp lại giữa các bước khác nhau của quá trình.

Chẳng hạn sau cuộc họp sơ bộ, người ta bắt đầu quá trình bình thường của mình cho đến khi người ta ở trong giai đoạn phân tích dữ liệu khám phá. Thực tế cho thấy sự khác biệt trong hành vi giữa hai nhóm, có thể khác biệt do nam, nữ. Cần lại truy cập dữ liệu để xác nhận điều này. Đối với điều này, cần phải trải qua quá trình phê duyệt, điều này cho thấy rằng người ta, hoặc doanh nghiệp, cần cung cấp một loại *qui định*<sup>1</sup> dự án. Ở các công ty lớn, việc lấy tất cả dữ liệu cần thiết để hoàn thành dự án có thể là một thử thách.

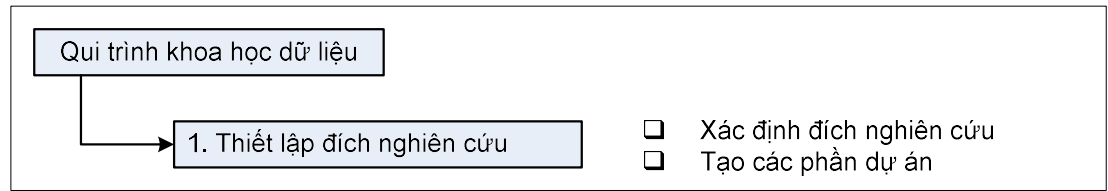
### 2.2.2. Xác định mục tiêu nghiên cứu và tạo qui định dự án

Một dự án bắt đầu bằng cách *hiểu* những gì, tại sao và như thế nào. Một số câu hỏi đặt ra như (i) công ty mong đợi gì?; (ii) lí do ban quản lí lại coi trọng nghiên cứu như vậy? (iii) đây là một phần của bức tranh chiến lược lớn hơn hay một dự án đơn lẻ, bắt nguồn từ một cơ hội mà ai đó phát hiện? Trả lời ba câu hỏi này, tức câu hỏi về *cái gì, tại sao, như thế nào*, là mục tiêu của giai đoạn đầu tiên, để mọi người biết phải làm gì và có thể thống nhất về hướng hành động tốt nhất.

*Kết quả* phải là (i) một mục tiêu nghiên cứu rõ ràng; (ii) hiểu biết tốt về bối cảnh; (iii) các công việc được xác định rõ ràng; (iv) một kế hoạch hành động có thời gian biểu. Thông tin này tốt nhất được đặt trong qui định dự án. Tất nhiên, dung lượng và hình thức của qui định có thể khác nhau giữa các dự án và công ty. Trong giai đoạn đầu của dự án, kĩ năng con người và sự nhạy bén trong kinh doanh quan trọng hơn năng lực kĩ thuật, đó là lí do phần này thường sẽ được nhân sự cấp cao hơn dẫn dắt.

---

<sup>1</sup> Charter: qui định, điều lệ.



**Hình 2.4. Thiết lập đích nghiên cứu**

#### 2.2.2.1. Dành thời gian tìm hiểu đích và ngữ cảnh nghiên cứu

Một kết quả thiết yếu là *đích*<sup>1</sup> nghiên cứu, nêu rõ mục đích của nhiệm vụ một cách rõ ràng và tập trung. Hiểu được các đích, tức mục tiêu, và *ngữ cảnh*<sup>2</sup> kinh doanh là rất quan trọng cho sự thành công của dự án. Tiếp tục đặt câu hỏi và đưa ra các thí dụ cho đến khi (i) nắm được kỳ vọng kinh doanh chính xác; (ii) xác định cách dự án phù hợp với bức tranh lớn hơn; (iii) cách để nghiên cứu sẽ thay đổi doanh nghiệp; (iv) cách công ty sẽ sử dụng kết quả. Khi báo cáo lại phát hiện của mình cho tổ chức, có thể mọi người ngay lập tức nhận ra sai lầm đâu đó. Không nên lướt qua giai đoạn này một cách nhẹ nhàng. Nhiều nhà khoa học dữ liệu thất bại ở đây: ỷ lại vào thông minh về toán học và tài năng khoa học, họ dường như không bao giờ nắm bắt được các mục tiêu và ngữ cảnh kinh doanh.

#### 2.2.2.2. Tạo qui định của dự án

Khách hàng muốn biết trước những gì họ đang chi trả, vì vậy sau khi đã hiểu rõ về vấn đề kinh doanh, hãy cố gắng đạt được thỏa thuận chính thức về các sản phẩm được giao. Tất cả thông tin này được thu thập tốt nhất trong qui định của dự án. Đối với bất kỳ dự án quan trọng nào, điều này sẽ là bắt buộc.

Điều lệ dự án yêu cầu *làm việc theo nhóm* và đầu vào bao gồm ít nhất những điều sau:

- Mục tiêu nghiên cứu rõ ràng;
- Nhiệm vụ và ngữ cảnh của dự án;
- Cách thực hiện phân tích;
- Những tài nguyên muốn sử dụng;
- Bằng chứng cho phép khẳng định rằng dự án có thể đạt được hoặc bằng chứng về các khái niệm;
- Sản phẩm phân phối và thước đo thành công;
- Lịch trình về thời gian.

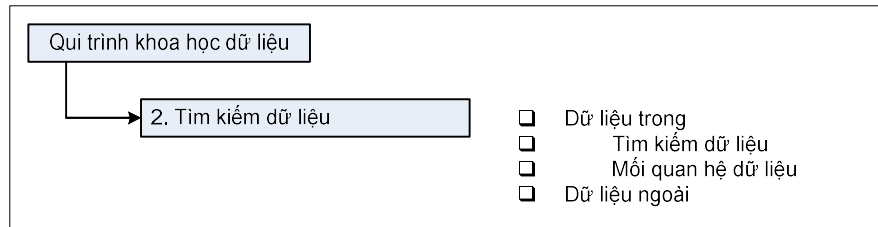
Khách hàng có thể sử dụng thông tin này để ước tính chi phí dự án, dữ liệu và con người cần thiết để dự án thành công.

<sup>1</sup> Goal: đích, mục tiêu cần đến.

<sup>2</sup> Context: ngữ cảnh, bối cảnh.

### 2.2.3. Tìm kiếm dữ liệu

Bước tiếp theo trong khoa học dữ liệu là tìm kiếm dữ liệu cần thiết. Đôi khi cần phải đi sâu vào thực địa và tự thiết kế quá trình thu thập dữ liệu, nhưng phần lớn thời gian của chuyên gia công nghệ thông tin sẽ không tham gia vào bước này. Nhiều công ty đã thu thập và lưu trữ dữ liệu và những gì họ không có, thường có thể được mua từ bên thứ ba. Đừng ngại tìm kiếm dữ liệu bên ngoài tổ chức, vì ngày càng có nhiều tổ chức cung cấp dữ liệu chất lượng cao miễn phí cho mục đích sử dụng công khai hay thương mại.



**Hình 2.5. Tìm kiếm dữ liệu**

Dữ liệu có thể được lưu trữ dưới nhiều hình thức, từ các (i) tệp văn bản đơn giản; đến (ii) các bảng dữ liệu trong cơ sở dữ liệu. Mục tiêu bây giờ là thu thập tất cả dữ liệu cần thiết. Điều này có thể khó, và ngay cả khi thành công, dữ liệu thường vẫn là dữ liệu thô; nó cần được làm tinh, làm sạch... để có ích hơn.

#### 2.2.3.1. Dữ liệu đã lưu trữ trong công ty

Hành động đầu tiên phải là đánh giá mức độ liên quan và chất lượng của dữ liệu sẵn có trong công ty. Hầu hết các công ty đều có một chương trình để duy trì dữ liệu quan trọng, vì vậy nhiều công việc dọn dẹp có thể đã được thực hiện. Dữ liệu này có thể được lưu trữ trong các kho dữ liệu chính thức như *cơ sở dữ liệu*, *kho dữ liệu chuyên đề*<sup>1</sup>, kho dữ liệu... do một nhóm chuyên gia công nghệ thông tin duy trì. Mục tiêu chính của cơ sở dữ liệu là lưu trữ dữ liệu, trong khi kho dữ liệu được thiết kế để đọc và phân tích dữ liệu đó. Kho dữ liệu chuyên đề là một tập hợp con của kho dữ liệu và hướng tới việc phục vụ một đơn vị kinh doanh cụ thể. Trong khi kho dữ liệu và kho dữ liệu chuyên đề là nơi chứa dữ liệu đã được xử lý trước, còn có các dữ liệu ở *dạng thô*. Có khả năng dữ liệu vẫn nằm trong các tệp Excel trên màn hình của một chuyên gia lĩnh vực.

Việc tìm kiếm dữ liệu ngay cả trong công ty đôi khi có thể là một thách thức. Khi các công ty phát triển, dữ liệu của họ trở nên phân tán ở nhiều nơi. Kiến thức về dữ liệu có thể bị phân tán khi mọi người thay đổi vị trí hay rời khỏi công ty. Tài liệu và *siêu dữ liệu*<sup>2</sup> không phải lúc nào cũng là ưu tiên hàng đầu đối với người quản lý phân phối, nên cần có biện pháp tìm ra dữ liệu đã mất. Truy cập vào dữ liệu là một

<sup>1</sup> Datamart: kho dữ liệu chuyên đề.

<sup>2</sup> Metadata: siêu dữ liệu, dữ liệu về dữ liệu.

nhiệm vụ khó khăn khác. Các tổ chức hiểu (i) giá trị; (ii) tính nhạy cảm của dữ liệu và thường có các chính sách để mọi người đều chỉ có quyền truy cập vào những gì họ cần. Các chính sách này chuyển thành các rào cản vật lí, như các *bức tường số*. Những bức tường này là bắt buộc và được quản lí chặt chẽ đối với dữ liệu khách hàng ở hầu hết các quốc gia. Điều này cũng vì những lí do chính đáng; hãy tưởng tượng mọi người trong một công ty thể tin dụng có quyền truy cập vào thói quen chi tiêu của con người. Việc truy cập vào dữ liệu có thể mất thời gian và liên quan đến chính sách của công ty.

#### *2.2.3.2. Đừng ngại mua sắm thêm*

Nếu dữ liệu không có sẵn trong tổ chức, hãy nhìn ra bên ngoài bức tường của tổ chức. Nhiều công ty chuyên thu thập thông tin có giá trị. Thí dụ, Nielsen và GfK nổi tiếng về điều này trong ngành bán lẻ. Các công ty khác cung cấp dữ liệu để người ta có thể làm phong phú thêm các dịch vụ và hệ sinh thái của họ. Đó là trường hợp của Twitter, LinkedIn và Facebook.

Mặc dù dữ liệu được một số công ty coi là tài sản quý giá, nhưng ngày càng có nhiều chính phủ và tổ chức chia sẻ dữ liệu của họ miễn phí với thế giới. Dữ liệu này có thể có chất lượng; nó phụ thuộc vào tổ chức tạo ra và quản lí nó. Thông tin họ chia sẻ bao gồm một loạt các chủ đề như (i) vụ tai nạn; (ii) lạm dụng ma túy; (iii) nhân khẩu học của khu vực địa lí. Dữ liệu này hữu ích khi người ta muốn làm giàu dữ liệu độc quyền và cũng thuận tiện khi đào tạo kĩ năng khoa học dữ liệu.

Danh sách các nhà cung cấp mã nguồn mở (i) *data.gov*, gồm các dữ liệu mở của chính phủ Hoa Kỳ; (ii) *freebase.org*, cơ sở dữ liệu mở, cho phép tìm kiếm thông tin từ các trang như Wikipedia, MusicBrains...; (iii) *data.worldbank.org*, dữ liệu mở cơ bản của ngân hàng thế giới; (iv) *aiddata.org*, dữ liệu mở đối với phát triển quốc tế...

#### *2.2.3.3. Kiểm tra chất lượng dữ liệu*

Kiểm tra chất lượng dữ liệu ngay bây giờ để ngăn ngừa các vấn đề có thể phát sinh. Dự kiến sẽ dành một phần lớn thời gian dự án để sửa và làm sạch dữ liệu, đôi khi lên đến 80%. Việc tìm kiếm dữ liệu là lần đầu tiên người ta kiểm tra dữ liệu trong quá trình khoa học dữ liệu. Hầu hết các lỗi người ta sẽ gặp phải trong giai đoạn thu thập dữ liệu đều dễ dàng phát hiện, nhưng nếu quá bất cẩn sẽ khiến mất nhiều giờ để giải quyết các vấn đề dữ liệu, mà có thể đã được ngăn chặn trong quá trình nhập dữ liệu.

Người ta sẽ *điều tra dữ liệu* trong giai đoạn nhập, chuẩn bị dữ liệu và khám phá. Sự khác biệt là ở mục tiêu và độ sâu của cuộc điều tra. Trong quá trình tìm kiếm dữ liệu, nên kiểm tra xem dữ liệu có bằng với dữ liệu trong tài liệu nguồn hay

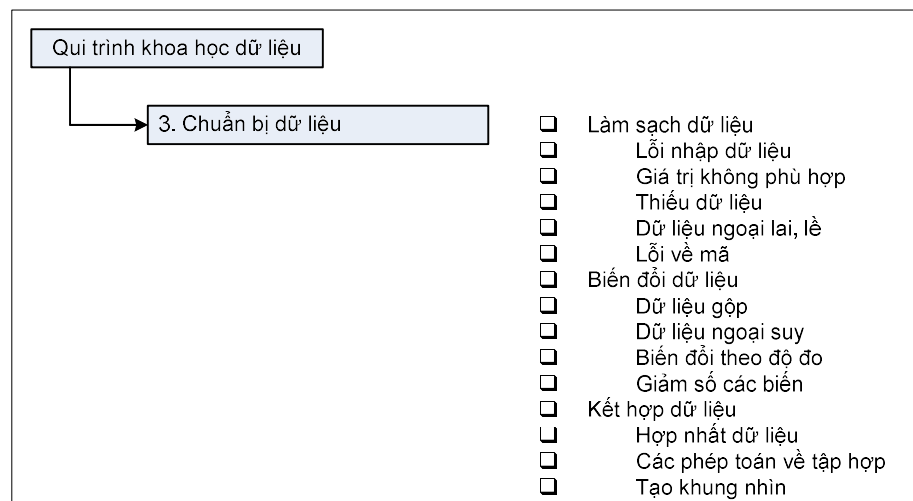


không và xem liệu có đúng kiểu dữ liệu hay không. Việc này sẽ không mất quá nhiều thời gian; khi người ta có đủ bằng chứng cho thấy dữ liệu tương tự với dữ liệu tìm thấy trong tài liệu nguồn, người ta sẽ dừng lại.

Việc chuẩn bị dữ liệu được thực hiện tỉ mỉ hơn. Nếu đã làm tốt công việc trong giai đoạn trước, các lỗi tìm thấy bây giờ cũng có trong tài liệu nguồn. Trọng tâm là nội dung của các biến: người ta muốn loại bỏ lỗi chính tả và các lỗi nhập dữ liệu khác và đưa dữ liệu về một tiêu chuẩn chung giữa các tập dữ liệu. Thí dụ có thể sửa *TW* thành *Trung ương*... Trong giai đoạn khám phá, trọng tâm chuyển sang những gì có thể học được từ dữ liệu. Bây giờ giả sử dữ liệu là sạch và xem xét các thuộc tính thống kê như (i) phân phối; (ii) tương quan; (iii) ngoại lệ. Cần thường xuyên lặp lại các giai đoạn này. Thí dụ: khi phát hiện ra các ngoại lệ trong giai đoạn khám phá, chúng có thể chỉ ra lỗi nhập dữ liệu.

#### 2.2.4. Làm sạch, tích hợp và chuyển đổi dữ liệu

Dữ liệu nhận được từ giai đoạn tìm kiếm dữ liệu có thể là thô. Nhiệm vụ tiếp là *làm sạch* và chuẩn bị nó để sử dụng trong giai đoạn lập mô hình và báo cáo. Làm như vậy là cực kỳ quan trọng vì (i) mô hình sẽ hoạt động tốt hơn; (ii) sẽ mất ít thời gian hơn khi cố gắng sửa đầu ra lại. Mô hình cần dữ liệu ở một định dạng cụ thể, do đó, việc *chuyển đổi* dữ liệu sẽ luôn hoạt động. Một thói quen tốt là sửa lỗi dữ liệu càng sớm càng tốt trong quá trình. Tuy nhiên, điều này không phải lúc nào cũng khả thi trong bối cảnh thực tế, vì vậy, sẽ cần thực hiện các hành động khắc phục trong chương trình.



**Hình 2.6. Chuẩn bị dữ liệu**

Hình 2.6 cho thấy các hành động phổ biến nhất cần thực hiện trong giai đoạn làm sạch, tích hợp và chuyển đổi dữ liệu. Bản đồ tư duy này hiện tại có thể hơi trừu tượng, nhưng người ta sẽ xử lý tất cả những điểm này một cách chi tiết hơn.

#### 2.2.4.1. Làm sạch dữ liệu

Làm sạch dữ liệu là một quá trình con của quá trình khoa học dữ liệu, tập trung vào việc loại bỏ các lỗi trong dữ liệu, để dữ liệu trở thành bản trình bày trung thực và nhất quán của các quá trình mà đã sinh ra dữ liệu. Bằng cách *thể hiện đúng và nhất quán*, thuật ngữ ngụ ý rằng có ít nhất hai loại lỗi. (i) Loại đầu tiên là lỗi diễn giải, chẳng hạn như khi điền sai giá trị trong dữ liệu của mình, chẳng hạn như nói rằng tuổi của một người lớn hơn 300 tuổi; (ii) Loại lỗi thứ hai chỉ ra sự mâu thuẫn giữa các nguồn dữ liệu hoặc so với các giá trị được chuẩn hóa của công ty. Một thí dụ về loại lỗi này là đặt *Nữ* vào một bảng và *F* trong một bảng khác khi chúng đại diện cho giới tính.

**Bảng 2.2. Tổng quan về các lỗi phổ biến**

Giải pháp tổng quát	
Cố gắng xác định sớm vấn đề khi thu thập dữ liệu hay xác định lỗi khi chạy chương trình	
Mô tả lỗi	Giải pháp đề xuất
Các lỗi gắn với các giá trị sai trong tập dữ liệu	
Lỗi khi nhập dữ liệu	Kiểm tra thủ công
Thừa khoảng trống	Sử dụng tự động thay thế trong văn bản
Các giá trị không được chấp nhận	Kiểm tra thủ công
Thiếu giá trị	Loại bỏ quan sát hay mất giá trị
Dữ liệu ngoại lai, lè	Kiểm tra sự hợp lí <sup>1</sup> . Nếu dữ liệu sai, có thể xem nó như dữ liệu thiếu.
Các lỗi dẫn đến mất toàn vẹn dữ liệu	
Suy diễn từ tài liệu	Tìm lại khóa của tài liệu, hoặc xử lý thủ công
Khác về đơn vị hay độ đo	Tính toán lại, chuyển đổi
Mức gộp lớn khác nhau	Đưa về cùng mức trừu tượng hóa



**Hình. Quá trình làm sạch dữ liệu<sup>2</sup>**

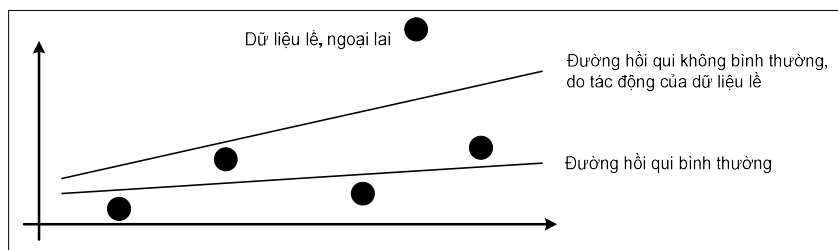
<sup>1</sup> Valid: hợp lí, đúng đắn. Phân biệt với verification, kiểm chứng, tức so sánh với giá trị đúng khác.

<sup>2</sup> Cleansing: làm sạch; Cleaning : phần nhỏ trong đó.



**Hình. Chu trình làm sạch**

Đôi khi, người ta sử dụng các phương pháp mạnh hơn, chẳng hạn như mô hình hóa đơn giản, để tìm và xác định lỗi dữ liệu; các chẩn đoán có thể đặc biệt có ý nghĩa. Thí dụ, trong hình 2.7 người ta sử dụng đồ họa để xác định các điểm dữ liệu có vẻ không đúng vị trí. Người ta thực hiện hồi quy để làm quen với dữ liệu và phát hiện ảnh hưởng của các quan sát riêng lẻ lên đường hồi quy. Khi một quan sát đơn lẻ có quá nhiều ảnh hưởng, điều này có thể chỉ ra lỗi trong dữ liệu, nhưng nó cũng có thể là một điểm hợp lệ. Tuy nhiên, ở giai đoạn làm sạch dữ liệu, các phương pháp tiên tiến này hiếm khi được áp dụng và thường được các nhà khoa học dữ liệu coi là quá mức cần thiết.



**Hình 2.7. Ảnh hưởng của dữ liệu ngoại lai**

### *Lỗi nhập dữ liệu*

Thu thập dữ liệu và nhập dữ liệu là những quá trình dễ xảy ra lỗi. Chúng thường yêu cầu sự can thiệp của con người, nên thường có lỗi chính tả hoặc mất tập trung trong một giây và đưa ra một lỗi trong chuỗi. Nhưng dữ liệu do máy móc hoặc máy tính thu thập cũng không tránh khỏi lỗi. Lỗi có thể phát sinh từ sự cẩu thả của con người, trong khi những lỗi khác là do lỗi máy hoặc phần cứng. Thí dụ về lỗi bắt nguồn từ máy móc là lỗi đường truyền hoặc lỗi trong giai đoạn trích xuất, biến đổi và tải dữ liệu ETL<sup>1</sup>.

<sup>1</sup> ETL: Extract - Transform – Load: trích xuất, biến đổi, tải.

Đối với các tập dữ liệu nhỏ, có thể kiểm tra thủ công mọi giá trị. Việc phát hiện lỗi dữ liệu khi các biến nghiên cứu không có nhiều lớp có thể được thực hiện bằng cách lập bảng dữ liệu với số lượng. Khi có một biến chỉ có thể nhận hai giá trị (i) tốt; (ii) xấu, người ta có thể tạo bảng tần suất và xem liệu đó có thực sự là hai giá trị duy nhất hiện có hay không.

### *Khoảng trắng dư thừa*

*Khoảng trắng*<sup>1</sup> có xu hướng khó phát hiện nhưng lại gây ra lỗi giống như các ký tự thừa khác. Người ta thường gặp lỗi về thừa khoảng trống trong dữ liệu văn bản, khiến việc so khớp các đoạn văn bản trở nên không chính xác. Người ta yêu cầu chương trình nối hai khóa và nhận thấy rằng các quan sát bị thiếu trong tệp đầu ra. Sau nhiều ngày tìm kiếm thông qua mã, cuối cùng người ta cũng tìm thấy lỗi. Do không làm tốt quá trình làm sạch trong giai đoạn ETL, các khóa trong một bảng chứa khoảng trắng ở cuối chuỗi. Điều này gây ra sự không khớp của các khóa, làm giảm các quan sát không khớp. Nếu biết để ý chúng, việc sửa các khoảng trắng dư thừa rất dễ dàng trong hầu hết các ngôn ngữ lập trình. Tất cả chúng đều cung cấp các hàm chuỗi sẽ loại bỏ các khoảng trắng ở đầu và cuối.

### *Xử lý việc không khớp khi so sánh ký tự viết hoa*

Không trùng khớp về chữ hoa là phổ biến. Hầu hết các ngôn ngữ lập trình phân biệt giữa *Hanoi* với *hanoi*. Trong trường hợp này, có thể giải quyết vấn đề bằng cách áp dụng một hàm trả về cả hai chuỗi bằng chữ thường, chẳng hạn như *lower()* trong Python.

### *Các giá trị không thể và kiểm tra tính đúng*

Kiểm tra tình trạng dữ liệu là một loại kiểm tra đặc biệt. Cần kiểm tra giá trị của biến so với các giá trị không thể thực hiện được về mặt vật lý hoặc lý thuyết, chẳng hạn như người cao hơn 3 mét hoặc người nào đó có tuổi 299. Kiểm tra *tính đúng*<sup>2</sup> tình trạng có thể được thể hiện trực tiếp với các điều kiện về giá trị.

### *Ngoại lệ*

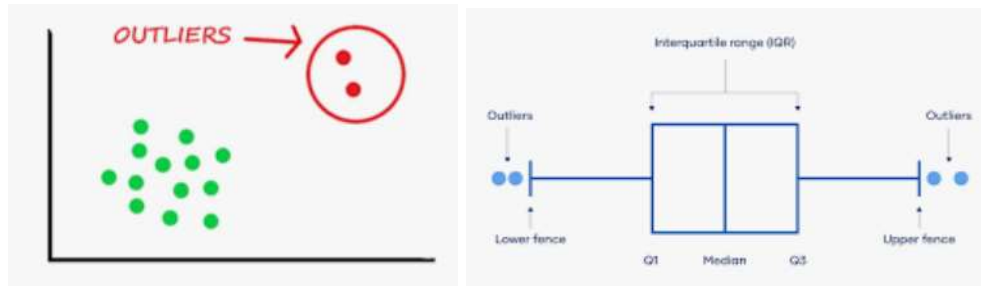
Một quan sát *ngoại lệ*, *lề*<sup>3</sup> là một quan sát dường như khác xa với các quan sát khác hoặc cụ thể hơn là một quan sát tuân theo một quá trình logic hoặc quá trình tổng hợp khác với các quan sát khác. Cách dễ nhất để tìm ra các giá trị ngoại lệ là sử dụng một biểu đồ hoặc một bảng có các giá trị tối thiểu và lớn nhất. Một thí dụ được thể hiện trong hình 2.8.

---

<sup>1</sup> Whitespace: khoảng trắng.

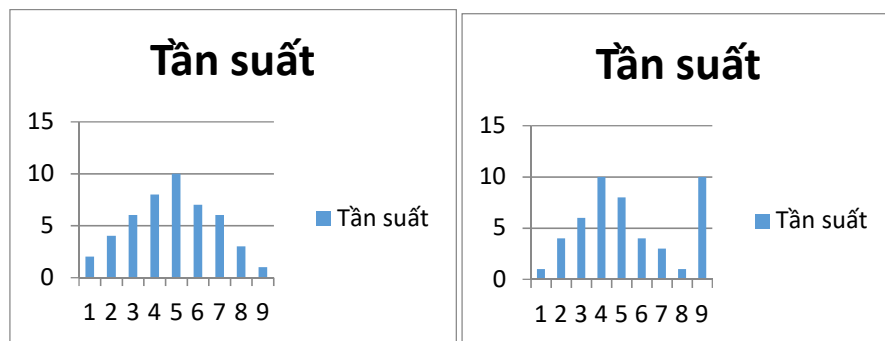
<sup>2</sup> Sanity: tính sạch, đúng đắn.

<sup>3</sup> Outlier: lề, ngoại lệ.



**Hình. Ngoại lệ**

Biểu đồ ở bên trái không hiển thị ngoại lệ, trong khi biểu đồ ở bên phải hiển thị các ngoại lệ có thể có ở phía trên khi dự kiến có phân phối chuẩn. Phân phối chuẩn, hay phân phối Gauss, là phân phối phổ biến nhất trong khoa học tự nhiên.



**Hình 2.8. Dựa vào phân phối để kiểm tra ngoại lệ**

Nó cho thấy hầu hết các trường hợp xảy ra xung quanh mức trung bình của phân phối và số lần xuất hiện càng giảm khi càng xa mức trung bình. Các giá trị cao trong biểu đồ bên phải có thể chỉ ra các giá trị ngoại lệ khi giả định phân phối chuẩn. Như đã thấy trước đó với thí dụ hồi quy, các giá trị ngoại lai có thể ảnh hưởng nghiêm trọng đến mô hình dữ liệu, vì vậy hãy điều tra chúng trước.

### *Xử lý khi thiếu dữ liệu*

Các giá trị thiếu<sup>1</sup> không nhất thiết là sai, nhưng cần xử lý chúng một cách riêng biệt; các kỹ thuật mô hình nhất định không thể xử lý các giá trị bị thiếu. Chúng có thể là chỉ báo cho thấy (i) đã xảy ra lỗi trong quá trình thu thập dữ liệu; hoặc (ii) đã xảy ra lỗi trong quá trình ETL. Các kỹ thuật phổ biến mà các nhà khoa học sử dụng dữ liệu được liệt kê trong bảng 2.3.

**Bảng 2.3. Các kỹ thuật xử lý dữ liệu thiếu**

Kỹ thuật	Ưu điểm	Nhược điểm
Bỏ sót giá trị.	Dễ thực hiện.	Mất thông tin về một quan sát.
Lấy giá trị là rỗng.	Dễ thực hiện.	Không phải mô hình nào cũng có thể lấy giá trị rỗng.

<sup>1</sup> Missing value: giá trị thiếu.

Xem giá trị tính là 0 hay giá trị trung bình.	Dễ thực hiện. Không mất thông tin đối với các biến khác trong quan sát.	Có thể dẫn đến đánh giá sai đối với mô hình.
Lấy một giá trị được đánh giá hay giá trị từ phân phối lí thuyết.	Không ảnh hưởng nhiều đến mô hình	Khó thực hiện. Người ta cần giả định về dữ liệu.
Mô hình hóa một giá trị.	Không ảnh hưởng nhiều đến mô hình.	Có thể dẫn đến việc quá tin vào mô hình. Có thể tự tạo nên mối quan hệ giữa các biến. Khó thực hiện, cần có giả định về dữ liệu.

Xử lý trường hợp thiếu dữ liệu ra sao là tùy vào thời điểm và hoàn cảnh lấy dữ liệu. Nếu dữ liệu có phân bố chuẩn, người ta có thể ước lượng được giá trị thiếu. Tuy nhiên việc thiếu dữ liệu nhiều khi là do thực tế; khi đó thông tin về việc thiếu là có ý nghĩa.

### *Sử dụng sách mã*

Việc phát hiện lỗi trong các tập dữ liệu lớn, so với (i) *sách mã*<sup>1</sup>; (ii) giá trị đã chuẩn hóa, có thể được thực hiện với sự trợ giúp của các phép toán tập hợp. Sách mã là mô tả dữ liệu, một dạng siêu dữ liệu. Nó chứa những thứ như số lượng biến trên mỗi quan sát, số lượng quan sát và ý nghĩa của mỗi mã hóa trong một biến. Chẳng hạn trong mạch IoT, *Gnd* ứng với đất, *Vcc* ứng với nguồn 5v. Một cuốn sách mã cũng cho biết loại dữ liệu đang xem: nó có thứ bậc, biểu đồ, thứ gì khác không? Không phải ngẫu nhiên mà các tập hợp là cấu trúc dữ liệu mà người ta sử dụng khi làm việc trong mã. Nên suy nghĩ thêm về cấu trúc dữ liệu của mình; nó có thể tiết kiệm công việc và cải thiện hiệu suất của chương trình. Nếu có nhiều giá trị cần kiểm tra, tốt hơn nên đặt chúng từ sách mã vào một bảng và sử dụng toán tử trừ để kiểm tra sự khác biệt giữa cả hai bảng.

### *Các đơn vị đo khác nhau*

Khi tích hợp hai tập dữ liệu, phải chú ý đến các đơn vị đo lường tương ứng của chúng. Một thí dụ về điều này là nghiên cứu giá xăng trên thế giới. Để làm điều này, người ta thu thập dữ liệu từ các nhà cung cấp dữ liệu khác nhau. Các tập dữ liệu có thể chứa giá mỗi *gallon*<sup>2</sup> và các tập khác có thể chứa giá mỗi *lít*. Một chuyển đổi đơn giản sẽ thực hiện thủ thuật trong trường hợp này.

<sup>1</sup> Code book: sách mã.

<sup>2</sup> Gallon: đơn vị đo thể tích, chất lỏng. Gallon ứng với 4,54609 lít, được sử dụng ở Vương quốc Anh, Canada và một số quốc gia Caribe; gallon Hoa Kỳ, 231 inch khối, 3,785411784 lít, được sử dụng ở Mỹ và một số nước Mỹ Latinh và Caribe; gallon khô của Hoa Kỳ, 1/8 giá Mỹ, 4,40488377086 lít.

### *Các mức độ gộp khác nhau*

Các mức độ tổng hợp khác nhau cũng tương tự như việc có các kiểu đo lường khác nhau. Thí dụ về điều này sẽ là tập dữ liệu chứa dữ liệu mỗi tuần so với tập dữ liệu chứa dữ liệu mỗi tuần làm việc. Loại lỗi này thường dễ phát hiện và việc tóm tắt, hoặc đảo ngược, mở rộng, các tập dữ liệu sẽ khắc phục được lỗi đó.

Sau khi làm sạch các lỗi dữ liệu, người ta kết hợp thông tin từ các nguồn dữ liệu khác nhau.

### *2.2.4.2. Sửa lỗi càng sớm càng tốt*

Một kinh nghiệm là sớm xử lý các lỗi về dữ liệu trong tập dữ liệu, giảm thời gian sửa lỗi trong chương trình. Tìm kiếm dữ liệu là một nhiệm vụ khó khăn và các tổ chức mất nhiều chi phí cho việc tìm kiếm và thu thập dữ liệu. Quá trình thu thập dữ liệu là quá trình dễ sai sót, và trong một tổ chức lớn, nó bao gồm nhiều bước và liên quan đến nhiều nhóm.

Dữ liệu phải được làm sạch khi có được vì nhiều lý do:

- Không phải ai cũng phát hiện ra sự bất thường của dữ liệu. Người ra quyết định có thể mắc sai lầm tốn kém về thông tin dựa trên dữ liệu không chính xác từ các ứng dụng không sửa được dữ liệu bị lỗi;
- Nếu các lỗi không được sửa chữa sớm trong quá trình này, việc làm sạch phải được thực hiện cho mọi dự án sử dụng dữ liệu đó;
- Lỗi dữ liệu có thể chỉ ra một quá trình kinh doanh không hoạt động như thiết kế. Thí dụ, cả hai tác giả đều làm việc tại một cửa hàng bán lẻ trước đây và họ đã thiết kế một hệ thống chuyển đổi để thu hút nhiều người hơn và tạo ra lợi nhuận cao hơn. Trong một dự án khoa học dữ liệu, người ta đã phát hiện ra những khách hàng đã lạm dụng hệ thống chuyển tiền và kiếm tiền khi mua hàng tạp hóa. Mục tiêu của hệ thống phiếu khuyến mãi là để kích thích việc bán chéo, không phải cung cấp sản phẩm miễn phí. Lỗi hổng này khiến công ty phải trả giá và không ai trong công ty biết về nó. Trong trường hợp này, dữ liệu không sai về mặt kỹ thuật nhưng mang lại kết quả không mong đợi;
- Lỗi dữ liệu có thể chỉ đến thiết bị bị lỗi, chẳng hạn như đường truyền bị hỏng và cảm biến bị lỗi;
- Lỗi dữ liệu có thể chỉ ra các lỗi trong phần mềm hoặc trong việc tích hợp phần mềm, mà chúng quan trọng đối với công ty. Trong khi thực hiện một dự án nhỏ tại một ngân hàng, người ta phát hiện ra rằng hai ứng dụng phần mềm sử dụng các cài đặt cục bộ khác nhau. Điều này gây ra sự cố với các số lớn hơn 1.000. Đối với một ứng dụng, con số 1.000 có nghĩa là một và đối với ứng dụng khác, nó có nghĩa là một nghìn.



Việc sửa dữ liệu ngay sau khi thu thập là một điều tốt. Tuy nhiên, nhà khoa học dữ liệu không phải lúc nào cũng có tiếng nói trong việc thu thập dữ liệu và chỉ cần yêu cầu bộ phận công nghệ thông tin sửa một số điều nhất định. Nếu không thể sửa dữ liệu tại nguồn, người ta cần xử lý dữ liệu đó bên trong mã chương trình. Thao tác dữ liệu không kết thúc bằng việc sửa lỗi; người ta vẫn cần kiểm tra sự tương thích với các dữ liệu đã có sẵn, để có bộ dữ liệu tổng thể. Lưu ý cuối cùng: hãy luôn giữ một bản sao dữ liệu gốc, nếu có thể. Đôi khi, bắt đầu làm sạch dữ liệu nhưng sẽ mắc sai lầm: đưa vào các biến sai cách, xóa các giá trị ngoại lệ có thông tin bổ sung thú vị hoặc thay đổi dữ liệu do hiểu sai ban đầu. Nếu giữ một bản sao, người ta có thể thử lại. Đối với *dữ liệu luân chuyển* được thao tác tại thời điểm bắt đầu, điều này không phải lúc nào cũng có thể thực hiện được và sẽ chấp nhận một khoảng thời gian điều chỉnh trước khi sử dụng dữ liệu. Tuy nhiên, một trong những điều khó khăn hơn không phải là việc làm sạch dữ liệu của các tập dữ liệu riêng lẻ, mà là việc kết hợp các nguồn khác nhau thành bộ dữ liệu tổng thể có ý nghĩa hơn.

#### *2.2.4.3. Kết hợp dữ liệu từ các nguồn dữ liệu khác nhau*

Dữ liệu đến từ nhiều nơi khác nhau và nên đề cập việc tích hợp các nguồn khác nhau này. Dữ liệu khác nhau về kích thước, kiểu và cấu trúc, từ cơ sở dữ liệu và tệp Excel đến tài liệu văn bản.

Sau đây sẽ (i) tập trung vào dữ liệu trong các cấu trúc bảng, để ngắn gọn; (ii) chú trọng quá trình khoa học dữ liệu, thay vì trình bày các tình huống cho mọi loại dữ liệu. Nhưng cũng nên nhớ còn có các nguồn dữ liệu khác, chẳng hạn như (i) kho khóa-giá trị; (ii) kho tài liệu...

#### *Các cách kết hợp dữ liệu khác nhau*

Có thể thực hiện hai phép toán để kết hợp thông tin từ các tập dữ liệu khác nhau: (i) phép đầu tiên là *nối*<sup>1</sup>: làm phong phú thêm một quan sát từ một bảng với thông tin từ một bảng khác; (ii) phép thứ hai là *thêm* hoặc *xếp chồng*: thêm các quan sát của một bảng vào các quan sát của bảng khác.

Khi kết hợp dữ liệu, có thể (i) tạo bảng vật lý mới; hoặc (ii) sử dụng bảng ảo bằng cách tạo *khung nhìn*<sup>2</sup>. Ưu điểm của chế độ xem là nó không tiêu tốn nhiều dung lượng bộ nhớ.

#### *Các bảng nối*

Phép nối của đại số quan hệ cho phép nối hai bảng dữ liệu theo điều kiện nối. Tùy theo điều kiện nối mà người ta chia ra nhiều loại nối (i) nối bằng; (ii) nối  $\theta$ ;

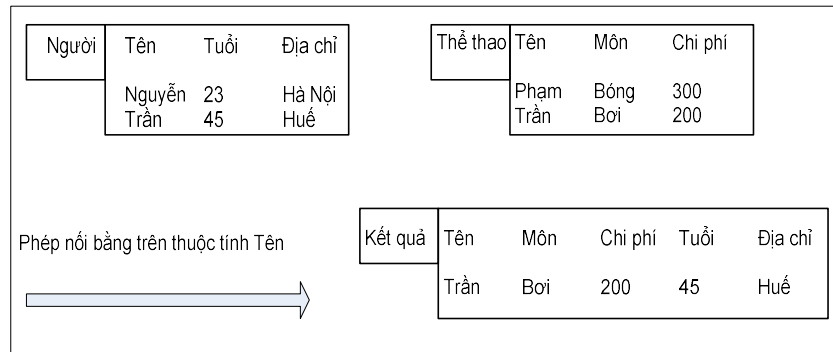
---

<sup>1</sup> Join: nối, kết nối. Phép nối là phép trong đại số quan hệ, là dẫn xuất của phép (i) tích Decac; và (ii) hạn chế.

<sup>2</sup> View: khung nhìn.



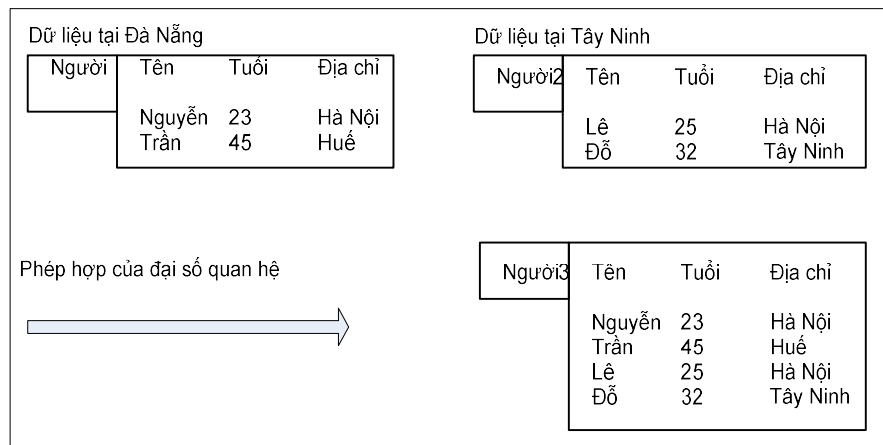
(iii) tự nối; (iv) nửa nối; (v) nối tự nhiên. Hình cho thấy bảng nối được tạo mới nhờ nối hai bảng theo khóa *Tên* của chúng.



**Hình 2.9. Nối hai bảng dữ liệu**

### *Bổ sung các bản ghi vào một bảng*

Bổ sung thêm dữ liệu, từ bảng này sang bảng khác, cũng nhằm tạo thêm thông tin cho bảng lớn. Trong hệ thống phân tán, các phép toán phân tán không nhất thiết yêu cầu tập hợp dữ liệu lại trong bảng lớn hơn. Người ta có bảng dữ liệu lớn hơn cũng nhằm tạo điều kiện thuận lợi cho quá trình ra quyết định.

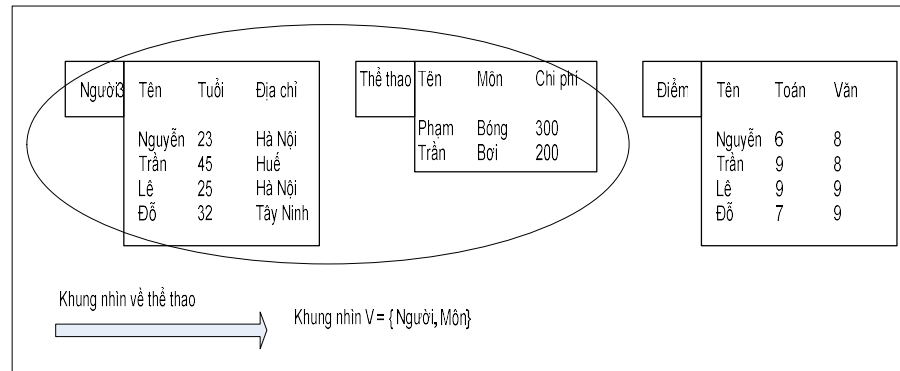


**Hình 2.10. Tích hợp các dữ liệu phân tán**

### *Sử dụng khung nhìn để mô phỏng phép toán nối và bổ sung dữ liệu*

Để tránh trùng lặp dữ liệu, hầu như người ta kết hợp dữ liệu với các khung nhìn. Có thể lấy dữ liệu hàng tháng và kết hợp nó trong một bảng vật lí mới; vậy đã sao chép dữ liệu và do đó cần thêm dung lượng lưu trữ. Tuy không có sự cố, nhưng hãy tưởng tượng rằng mọi bảng đều chứa hàng terabyte dữ liệu; thì việc sao chép dữ liệu sẽ trở thành vấn đề. Vì lí do này, khái niệm về một khung nhìn là cần thiết. Khung nhìn hoạt động như thể đang làm việc trên một bảng, nhưng bảng này không là gì ngoài một lớp ảo kết hợp các bảng khác. Hình 2.11 cho thấy cách dữ liệu bán hàng từ các tháng khác nhau được kết hợp ảo thành một bảng doanh số hàng năm

thay vì sao chép dữ liệu. Tuy nhiên, khung nhìn có một nhược điểm. Mặc dù phép nối bảng chỉ được thực hiện một lần, phép nối tạo khung nhìn được tạo lại mỗi khi nó được truy cập, sử dụng thời gian xử lý nhiều hơn so với tiếp cận sao chép.



**Hình 2.11. Khung nhìn cơ sở dữ liệu**

### *Làm giàu dữ liệu nhờ thông tin tích hợp*

Việc làm giàu dữ liệu cũng có thể được thực hiện bằng cách thêm thông tin đã tính toán vào bảng, chẳng hạn như tổng số lần bán hàng hoặc tỉ lệ phần trăm tổng số hàng đã được bán ở một khu vực nhất định. Các thông tin tích hợp như thế cho phép nhìn nhận thuận tiện hơn về thực tại. Lúc này người ta có một tập hợp dữ liệu tổng hợp, cho phép tính toán để biết sự tham gia của từng sản phẩm trong danh mục của nó. Điều này có thể hữu ích khi khám phá dữ liệu nhưng còn hơn thế nữa khi tạo mô hình dữ liệu. Như mọi khi, điều này phụ thuộc vào từng trường hợp chính xác, nhưng theo kinh nghiệm, các mô hình *đo tương đối* như *phần trăm* có xu hướng tốt hơn các mô hình sử dụng số liệu tuyệt đối làm đầu vào.

Dữ liệu thống kê				
Đơn vị	Sản xuất	Vượt mức	Phần trăm	Xếp hạng
Nhà máy dệt	300	50	0.17	4
Công trình làm đường	500	50	0.10	3
Sản xuất máy tính	6000	30	0.01	1
Xuất khẩu phần mềm	2000	500	0.25	2

Tỉ lệ tương đối so với tỉ lệ tuyệt đối về vượt mức

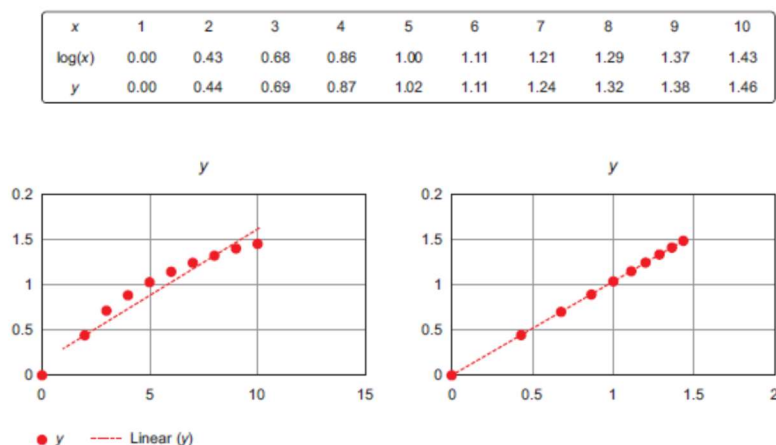
**Hình 2.12. Bảng dữ liệu thống kê trong phần mềm bảng tính Excel**

### *2.2.4.4. Chuyển đổi dữ liệu*

Một số mô hình yêu cầu dữ liệu của chúng phải có hình dạng nhất định. Sau khi dữ liệu được làm sạch và tích hợp, đây là nhiệm vụ tiếp theo về khoa học dữ liệu. Dữ liệu được chuyển đổi để có dạng phù hợp cho việc lập mô hình dữ liệu.

## Chuyển đổi dữ liệu

Mối quan hệ giữa biến đầu vào và biến đầu ra không phải lúc nào cũng tuyến tính. Lấy thí dụ, một mối quan hệ có dạng  $y = ae^{bx}$ . Lấy log của các biến độc lập sẽ đơn giản hóa vấn đề ước lượng một cách đáng kể. Hình 2.13 cho thấy cách biến đổi các biến đầu vào sẽ đơn giản hóa vấn đề ước lượng như thế nào. Những lần khác, có thể kết hợp hai biến thành một biến mới.



**Hình 2.13. Chuyển x thành logx sẽ tạo quan hệ tuyến tính giữa x và y, tốt hơn hình bên trái**

## Khoảng cách Ơclit

Khoảng cách Ơclit hay khoảng cách *chuẩn*, là một phần mở rộng đối với điều đầu tiên trong lượng giác: định lý Pytago. Nếu biết độ dài của hai cạnh góc 90° của một tam giác vuông, có thể dễ dàng suy ra độ dài của cạnh còn lại, tức cạnh huyền.

Công thức  $= \sqrt{(\text{cạnh}1)^2 + (\text{cạnh}2)^2}$ . Khoảng cách Ơclit giữa hai điểm trong mặt phẳng hai chiều được tính bằng công thức tương tự: khoảng cách

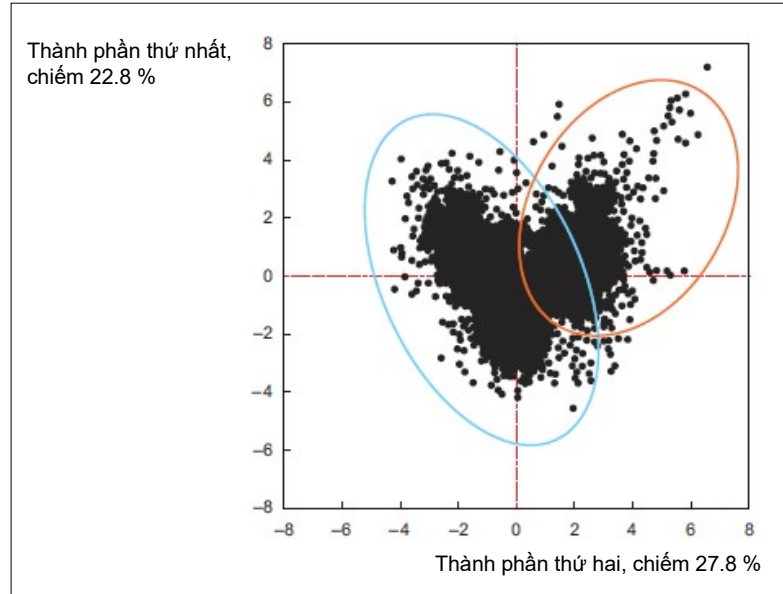
$= \sqrt{(x1 - x2)^2 + (y1 - y2)^2}$ . Nếu mở rộng phép tính khoảng cách này sang nhiều chiều hơn, hãy thêm tọa độ của điểm trong các kích thước cao hơn đó vào công thức. Chẳng hạn với ba chiều, khoảng cách

$$= \sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2}.$$

## Giảm số biến

Đôi khi có quá nhiều biến và cần giảm số lượng vì chúng không thêm thông tin cho mô hình. Có quá nhiều biến trong mô hình khiến mô hình khó xử lý và một số kỹ thuật không hoạt động tốt khi có quá nhiều biến đầu vào. Thí dụ, tất cả các kỹ thuật dựa trên khoảng cách Ơclit chỉ hoạt động tốt với tối đa 10 biến. Các nhà khoa học dữ liệu sử dụng các phương pháp đặc biệt để giảm số lượng biến nhưng giữ lại lượng dữ liệu tối đa.

Hình 2.14 cho thấy cách giảm số lượng biến giúp dễ hiểu các giá trị chính hơn. Nó cũng cho thấy hai biến chiếm 50,6% sự thay đổi trong tập dữ liệu, *thành phần1* = 27.8%, *thành phần2* = 22.8%. Các biến này đều là sự kết hợp của các biến ban đầu.



**Hình 2.14. Hai biến chiếm tỉ trọng lớn**

Chúng là thành phần chính của cấu trúc dữ liệu cơ bản. Để rõ thêm, có thể tham khảo nội dung về *phân tích các thành phần chính PCA*<sup>1</sup>. Những gì có thể thấy là sự hiện diện của một biến thứ ba, không xác định, chia tách nhóm quan sát thành hai.

### *Chuyển một biến thành biến câm*

Các biến có thể được chuyển thành *biến câm*<sup>2</sup>, hay biến giả. Biến giả chỉ có thể nhận hai giá trị: true, hay 1, hoặc false, hay 0. Chúng được sử dụng để biểu thị sự vắng mặt của hiệu ứng phân loại có thể giải thích cho việc quan sát. Trong trường hợp này, người ta sẽ tạo các cột riêng biệt cho các lớp được lưu trữ trong một biến và biểu thị nó bằng 1 nếu lớp có mặt và 0 nếu không. Một thí dụ là chuyển một cột có tên *Các ngày trong tuần* thành các cột từ *thứ hai* đến *chủ nhật*. người ta sử dụng một chỉ báo để biết liệu quan sát có vào *thứ hai* hay không; người ta đặt 1 vào *thứ hai*, 0 ở nơi khác. Biến các biến thành biến câm là một kĩ thuật được sử dụng trong mô hình hóa và phổ biến trong kinh tế học.

Phần này đã đề cập bước thứ ba trong quá trình khoa học dữ liệu, làm sạch, chuyển đổi và tích hợp dữ liệu; đã thay đổi dữ liệu thô thành dữ liệu đầu vào có thể

<sup>1</sup> PCA: Principal Components Analysis: phân tích thành phần chính.

<sup>2</sup> Dummy: biến câm, biến giả.

sử dụng cho giai đoạn mô hình hóa. Bước tiếp theo trong quá trình khoa học dữ liệu là hiểu rõ hơn về nội dung của dữ liệu và các mối quan hệ giữa các biến và các quan sát.

Khách hàng	Năm	Giới tính	Mua bán
Mơ	2018	Nữ	100
Mận	2020	Nam	20
Mơ	2021	Nữ	30
Đào	2020	Nam	200
Quất	2021	Nữ	50
Đào	2021	Nam	40

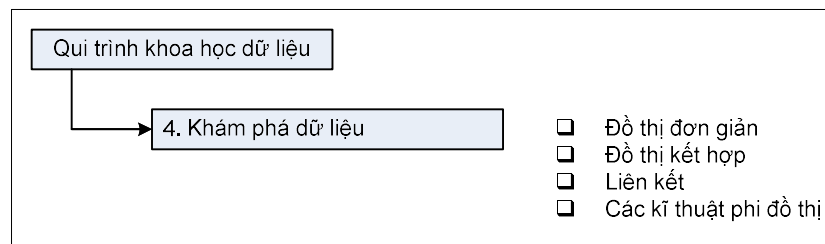
Tách thuộc tính Giới tính thành (i) nam; (ii) Nữ

Khách hàng	Năm	Nam	Nữ	Mua bán
Mơ	2018	0	1	100
Mận	2020	1	0	20
Mơ	2021	0	1	30
Đào	2020	1	0	200
Quất	2021	0	1	50
Đào	2021	1	0	40

**Hình 2.15. Sử dụng biến cam**

### 2.2.5. Phân tích dữ liệu khám phá

Trong quá trình phân tích dữ liệu khám phá, người ta đi sâu vào dữ liệu. Thông tin trở nên dễ nắm bắt hơn nhiều khi được hiển thị trong hình ảnh, do đó người ta chủ yếu sử dụng các kĩ thuật đồ họa để hiểu được dữ liệu của mình và sự tương tác giữa các biến. Giai đoạn này là về khám phá dữ liệu, do đó (i) cởi mở; (ii) quan sát là cần thiết. Đích không phải là làm sạch dữ liệu mà thông thường, mà là sẽ phát hiện ra những điểm bất thường đã bị bỏ qua trước đó, buộc người ta phải lùi lại một bước và sửa chúng.

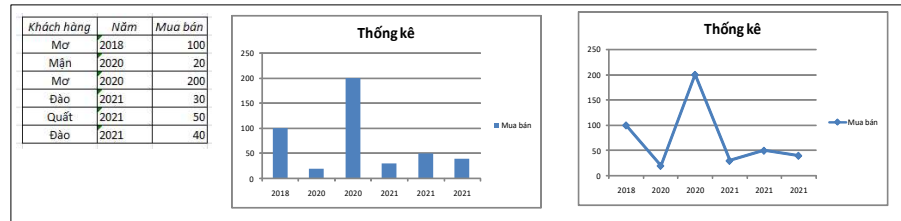


**Hình 2.16. Khám phá dữ liệu**

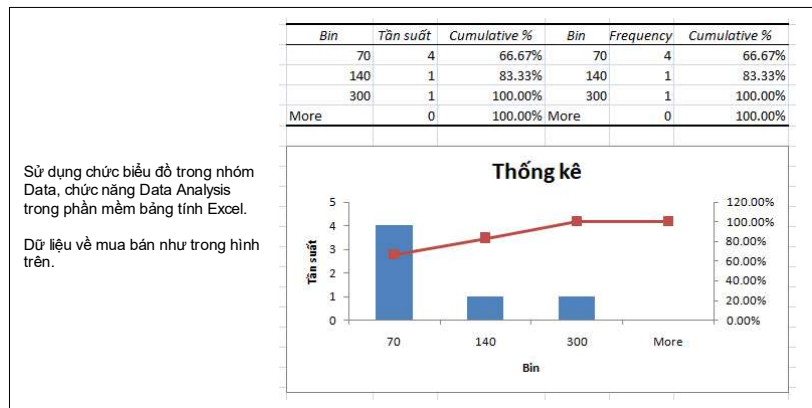
Các kĩ thuật trực quan hóa sử dụng trong giai đoạn này bao gồm từ (i) đồ thị đường đơn giản; (ii) *biểu đồ*<sup>1</sup>; đến (iii) các biểu đồ phức tạp hơn, như đồ thị Sankey

<sup>1</sup> Histogram: biểu đồ, thể hiện tần suất dạng cột.

và đồ thị mạng. Đôi khi, việc tạo một biểu đồ tổng hợp từ các biểu đồ đơn giản để hiểu sâu hơn về dữ liệu là việc cần thiết. Ngoài ra, các biểu đồ có thể được làm động hoặc tạo tương tác để làm cho nó dễ nhìn hơn.

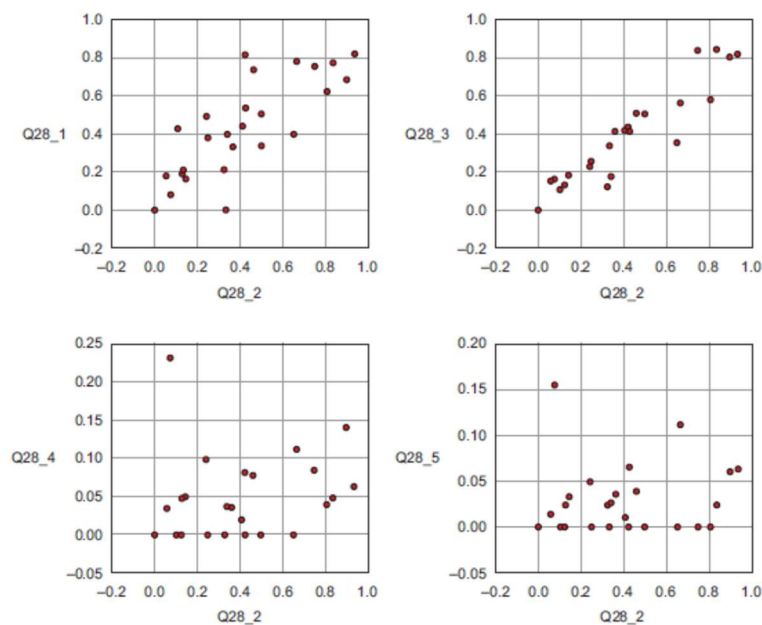


**Hình 2.17. Dữ liệu và đồ thị trong bảng tính Excel**



**Hình 2.18. Phân bố thống kê của dữ liệu**

Những *biểu đồ*<sup>1</sup> này có thể được kết hợp để cung cấp cái nhìn sâu sắc hơn, như thể hiện trong hình 2.18. Các biểu đồ chồng lên nhau là phổ biến.



<sup>1</sup> Plot: biểu đồ. Line plot: biểu đồ dạng đường.

### Hình 2.19. Đồ thị phức tạp hơn, mô tả cấu trúc của dữ liệu theo nhiều biến

Một kĩ thuật khác: *chải và liên kết*<sup>1</sup>. Với tính năng chải và liên kết, người ta kết hợp và liên kết các biểu đồ và bảng khác nhau để các thay đổi trong một biểu đồ được tự động chuyển sang các biểu đồ khác. Việc khám phá dữ liệu mang tính tương tác này tạo điều kiện thuận lợi cho việc khám phá những hiểu biết mới. Hình 2.19 cho thấy điểm trung bình của mỗi quốc gia cho các câu hỏi. Điều này không chỉ cho thấy mối tương quan cao giữa các câu trả lời mà còn dễ dàng nhận thấy rằng khi chọn một số điểm trên một biểu đồ con, các điểm đó sẽ tương ứng với các điểm tương tự trên các biểu đồ khác.

Hai biểu đồ quan trọng khác là (i) biểu đồ; (ii) biểu đồ hình hộp. Trong biểu đồ, một biến được cắt thành các danh mục riêng biệt và số lần xuất hiện trong mỗi danh mục được tổng hợp và hiển thị trong biểu đồ. Mặt khác, biểu đồ hình hộp không hiển thị số lượng quan sát hiện tại nhưng cung cấp ấn tượng về sự phân phối trong các danh mục. Nó có thể hiển thị các thước đo tối đa, tối thiểu, trung bình và các đặc điểm khác cùng một lúc.

Các kĩ thuật mô tả trong giai đoạn này chủ yếu là trực quan, nhưng trên thực tế, chúng chắc chắn không giới hạn ở các kĩ thuật trực quan. Lập bảng, phân cụm và các kĩ thuật mô hình hóa khác cũng có thể là một phần của phân tích khám phá. Ngay cả việc xây dựng các mô hình đơn giản cũng có thể là một phần của bước này.

#### 2.2.6. Công cụ tin học hóa

Cần thể hiện các quá trình khoa học dữ liệu trên máy tính, thông qua các phần mềm. Cũng có thể cứng hóa các xử lý, để có hệ thống phần cứng tự động thực hiện quá trình khoa học dữ liệu.

Liên quan đến công cụ cho khoa học dữ liệu, có (i) ngôn ngữ lập trình; (ii) môi trường cho phép tiện sử dụng các ngôn ngữ.

##### 2.2.6.1. Python và ngôn ngữ khoa học dữ liệu khác

Có thể sử dụng Python để thực hiện quá trình khoa học dữ liệu. Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

---

<sup>1</sup> Brushing and linking: chải và liên kết.





**Hình 2.20. Guido van Rossum, nhà sáng lập ra ngôn ngữ Python**

Python đơn giản. Tuy không hàn lâm như ngôn ngữ lập trình cấp cao Pascal, nó có thể được dùng như công cụ lập trình để giải vấn đề trong chương trình phổ thông.

```
1 """  
2 Created on Sep 9, 2017  
3  
4 @author: pattis  
5 """  
6 def cheer_up(times : int):  
7     for i in range(1,times+1):  
8         print('You will become a good programmer'+i*'!!')  
9  
10 sadness = input('Enter 1-10 indicating how sad you are: ')  
11 cheer_up(int(sadness))
```

Console output:  
<terminated> Script.py [C:\Users\Pattis\AppData\Local\Programs\Python\Python36-32\python.exe]  
Enter 1-10 indicating how sad you are: 5  
You will become a good programmer!  
You will become a good programmer!!

**Hình 2.21. Thí dụ một chương trình Python**

Các chuyên gia về khoa học dữ liệu cũng sử dụng (i) ngôn ngữ Julia; (ii) ngôn ngữ R.

- Julia là một ngôn ngữ lập trình có mục đích chung, đồng thời cũng được thiết kế cho tính toán số/kỹ thuật. Nó cũng hữu ích cho lập trình hệ thống cấp thấp, như một ngôn ngữ đặc tả và cho lập trình Web ở cả phía máy chủ và máy khách. Julia là một ngôn ngữ lập trình năng động, hiệu suất cao, cấp cao. Mặc dù nó là một ngôn ngữ có mục đích chung và có thể

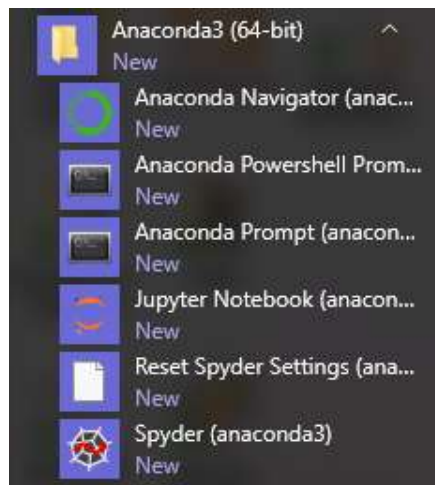


được sử dụng để viết bất kỳ ứng dụng nào, nhưng nhiều tính năng của nó rất phù hợp cho phân tích số và khoa học tính toán;

- R là một ngôn ngữ lập trình và môi trường phần mềm dành cho tính toán và đồ họa thống kê. R do Ross Ihaka và Robert Gentleman tạo ra tại Đại học Auckland, New Zealand. Nay nó được R Development Core Team chịu trách nhiệm phát triển. Tên của ngôn ngữ một phần lấy từ chữ cái đầu của hai tác giả, là Robert Gentleman và Ross Ihaka. Ngôn ngữ R đã trở thành một *chuẩn thực tế*<sup>1</sup> giữa các nhà thống kê, cho thấy sự phát triển của phần mềm thống kê, và được sử dụng rộng rãi để phát triển phần mềm thống kê và phân tích dữ liệu. R là một bộ phận của dự án *GNU*<sup>2</sup>. Mã nguồn của nó được công bố tự do theo Giấy phép Công cộng GNU, và có các phiên bản dịch sẵn cho nhiều hệ điều hành khác nhau. R sử dụng giao diện dòng lệnh, tuy cũng có một vài giao diện đồ họa người dùng dành cho nó.

#### 2.2.6.2. Công cụ Anaconda

Người dùng có thể sử dụng trực tiếp ngôn ngữ Python. Tuy nhiên để (i) có giao diện đồ họa; (ii) liên kết chương trình Python với các ngôn ngữ khác, người phát triển chương trình thường sử dụng Python trong môi trường của phần mềm khác. Trong các chương, Anaconda và Jupyter được dùng như môi trường cho Python.



**Hình 2.22. Anaconda được cài đặt trên máy tính**

*Conda* là một hệ thống quản lý môi trường và quản lý gói đa nền tảng. Ban đầu nó được phát triển để giải quyết những thách thức khó khăn về quản lý gói phần

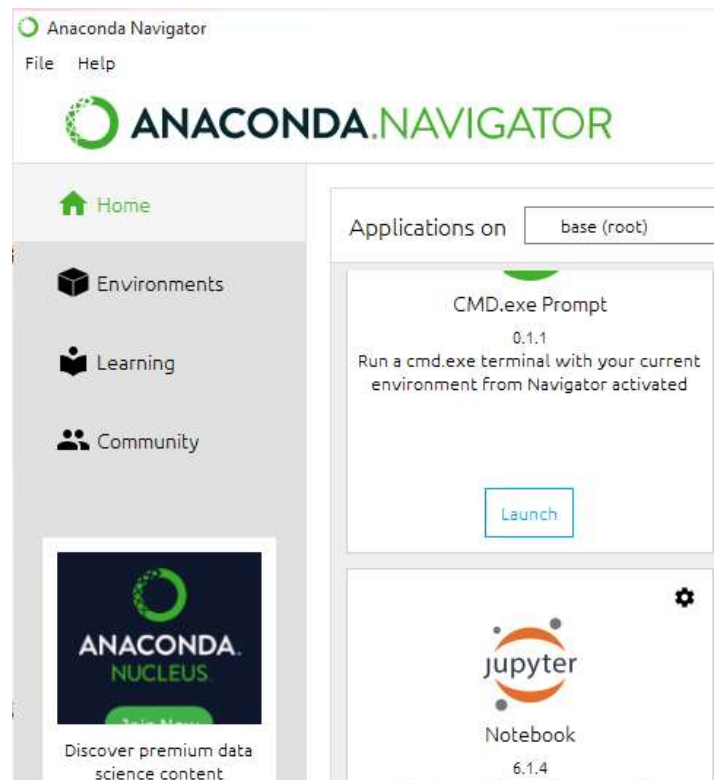
<sup>1</sup> De-facto: chuẩn thực tế.

<sup>2</sup> GNU: GNU's Not Unix: hệ điều hành và bộ sưu tập phần mềm máy tính phong phú.

mềm mà các nhà khoa học dữ liệu Python phải đối mặt và ngày nay là trình quản lí gói phổ biến cho Python và R.

*Anaconda* là một bản phân phối các ngôn ngữ lập trình Python và R cho tính toán khoa học, như (i) khoa học dữ liệu; (ii) ứng dụng học máy; (iii) xử lí dữ liệu qui mô lớn; (iv) phân tích dự đoán... nhằm mục đích đơn giản hóa việc quản lí và triển khai gói phần mềm. Bản phân phối bao gồm các gói phần mềm khoa học dữ liệu phù hợp với Windows, Linux và MacOS. Nó được phát triển và duy trì bởi Anaconda, Inc., do Peter Wang và Travis Oliphant thành lập vào năm 2012.

- *PyPI*: Chỉ mục gói Python, viết tắt là PyPI là kho phần mềm của bên thứ ba chính thức cho Python. Một số trình quản lí gói, bao gồm cả pip, sử dụng PyPI làm nguồn mặc định cho các gói và các phụ thuộc của chúng. Hơn 235.000 gói Python có thể được truy cập thông qua PyPI;
- *pip*<sup>1</sup> là một hệ thống quản lí gói được viết bằng Python được sử dụng để cài đặt và quản lí các gói phần mềm. Nó kết nối với một kho lưu trữ trực tuyến các gói riêng tư và công khai trả phí, được gọi là chỉ mục gói Python.



**Hình 2.23. Anacoda cho phép sử dụng Jupyter**

<sup>1</sup> Pip: Pip Installs Packages: gói cài đặt trong Python.

Phân phối Anaconda đi kèm với hơn 250 gói được cài đặt tự động và hơn 7.500 gói nguồn mở bổ sung có thể được cài đặt từ PyPI cũng như gói Conda và trình quản lý môi trường ảo. Nó cũng bao gồm GUI, Anaconda Navigator, như một sự thay thế đồ họa cho giao diện dòng lệnh.

Sự khác biệt lớn giữa Conda và trình quản lý gói pip là ở cách quản lý các gói phụ thuộc, đây là một thách thức đáng kể đối với khoa học dữ liệu Python và là lý do Conda tồn tại.

#### 2.2.6.3. Dự án Jupyter

Dự án *Jupyter*<sup>1</sup> là một tổ chức phi lợi nhuận được thành lập để phát triển phần mềm nguồn mở, các tiêu chuẩn mở và các dịch vụ cho máy tính tương tác trên hàng chục ngôn ngữ lập trình. Nó được tách ra từ IPython vào năm 2014. Dự án Jupyter trợ giúp các môi trường thực thi bằng hàng chục ngôn ngữ. Tên của dự án Jupyter liên quan đến ba ngôn ngữ lập trình cốt lõi được Jupyter trợ giúp, đó là Julia, Python và R.

Dự án Jupyter đã phát triển và trợ giúp các sản phẩm máy tính tương tác Jupyter Notebook, JupyterHub và JupyterLab.

#### 2.2.6.4. Sổ tay Jupyter

Giao diện *sổ ghi chép*<sup>2</sup> là một môi trường sổ ghi chép ảo được sử dụng để lập trình. Một sổ máy tính xách tay là môi trường WYSIWYG bao gồm các tính toán thực thi được nhúng trong các tài liệu được định dạng; những người khác tách các phép tính và văn bản thành các phần riêng biệt.

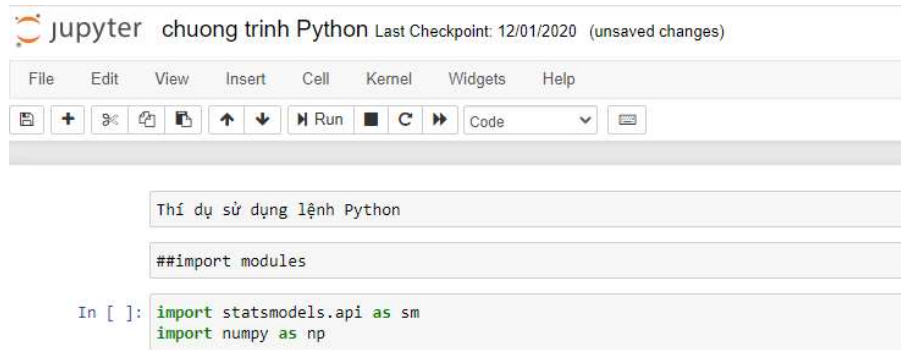
Sổ tay theo modul có thể kết nối với nhiều loại kết thúc tính toán khác nhau, được gọi là *nhân*. Giao diện sổ ghi chép được sử dụng rộng rãi cho thống kê, khoa học dữ liệu, học máy và đại số máy tính.

Giao diện sổ ghi chép được giới thiệu lần đầu tiên vào năm 1986 trong MathCad, trợ giúp các tính toán ký hiệu số, giới hạn. Khi giao diện sổ ghi chép ngày càng phổ biến trong hai thập kỷ tiếp theo, sổ ghi chép cho các đầu tính toán khác nhau, tức các nhân, đã được giới thiệu, bao gồm MATLAB, Python, Julia, Scala, SQL và các loại khác.

---

<sup>1</sup> Jupyter là tên ngầm định với ngôn ngữ (i) Python; (ii) Julia; (iii) R.

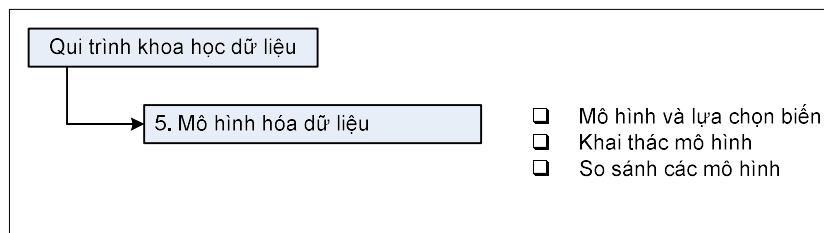
<sup>2</sup> Notebook: sổ ghi chép, sổ tay.



**Hình 2.24. Giao diện sổ tay Jupyter**

### 2.2.7. Xây dựng mô hình

Với dữ liệu rõ ràng và sự hiểu biết tốt về nội dung, người ta đã sẵn sàng xây dựng mô hình với mục tiêu đưa ra dự đoán tốt hơn, phân loại đối tượng hoặc hiểu rõ về hệ thống mà người ta tạo mô hình. Giai đoạn này tập trung hơn nhiều so với bước phân tích khám phá, vì người ta biết mình đang tìm kiếm điều gì và muốn kết quả như thế nào. Hình 2.25 cho thấy các thành phần của việc xây dựng mô hình.



**Hình 2.25. Quy trình xây dựng mô hình**

Các kỹ thuật sẽ sử dụng ở đây có xuất xứ từ lĩnh vực học máy, khai thác dữ liệu và/hoặc thống kê.

Xây dựng một mô hình là một quá trình lặp đi lặp lại. Cách xây dựng mô hình phụ thuộc vào (i) việc sử dụng thống kê cổ điển hay theo học máy; (ii) kỹ thuật muốn sử dụng. Dù bằng cách nào, hầu hết các mô hình đều bao gồm các bước chính sau:

1. Lựa chọn kỹ thuật mô hình hóa và các biến để nhập vào mô hình;
2. Thực hiện mô hình;
3. Chẩn đoán và so sánh mô hình.

#### 2.2.7.1. Mô hình và lựa chọn biến

Cần (i) chọn các biến muốn đưa vào mô hình; và (ii) kỹ thuật lập mô hình. Những phát hiện từ phân tích khám phá sẽ cung cấp ý tưởng hợp lý về những biến nào sẽ giúp xây dựng một mô hình tốt. Nhiều kỹ thuật lập mô hình có sẵn và việc lựa chọn mô hình phù hợp cho một vấn đề đòi hỏi sự phán đoán của người dùng. Cần phải xem xét hiệu suất mô hình và liệu dự án có đáp ứng tất cả các yêu cầu để sử

dựng mô hình hay không, rồi chuyển mô hình sang môi trường tạo nên; môi trường có tạo điều kiện để thực hiện không?

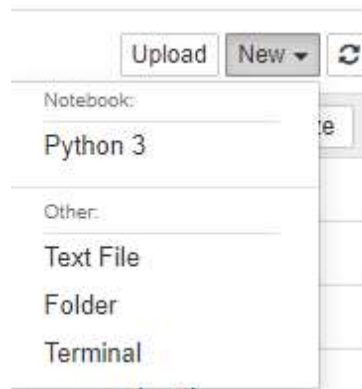
- Mức độ khó khăn của việc bảo trì mô hình: nó sẽ còn phù hợp trong bao lâu?
- Có nhu cầu về dễ giải thích mô hình không?

#### 2.2.7.2. Thực hiện mô hình

Khi đã chọn một mô hình, người ta sẽ cần triển khai mô hình đó. Việc này thể hiện với mã của ngôn ngữ lập trình; ở đây là Python.

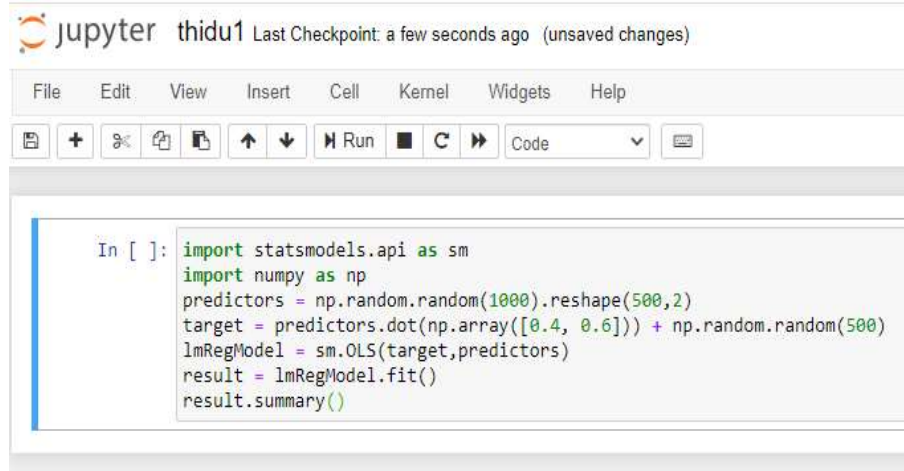
Mã hóa một mô hình là một nhiệm vụ không hề nhỏ trong hầu hết các trường hợp, vì vậy các thư viện có sẵn sẽ giúp đẩy nhanh quá trình. Như có thể thấy trong đoạn mã sau, khá dễ dàng để sử dụng hồi qui tuyến tính với *StatsModels* hoặc *Scikit-learning*. Tự làm điều này sẽ đòi hỏi nhiều nỗ lực hơn, ngay cả với kĩ thuật đơn giản. Đoạn mã chương trình sau đây cho thấy việc thực thi mô hình dự đoán tuyến tính.

Trước tiên, mở sổ tay Jupyter, tạo tệp mới, đặt tên. Chẳng hạn là *thidu1*. Sử dụng các lệnh Python. Ngôn ngữ trong sổ tay được chọn là Python.



**Hình 2.26. Bắt đầu với chọn tệp sổ tay mới, với loại Python3**

Nhiệm vụ của chương trình này là khai thác mô hình dự đoán tuyến tính, với dữ liệu nửa ngẫu nhiên.

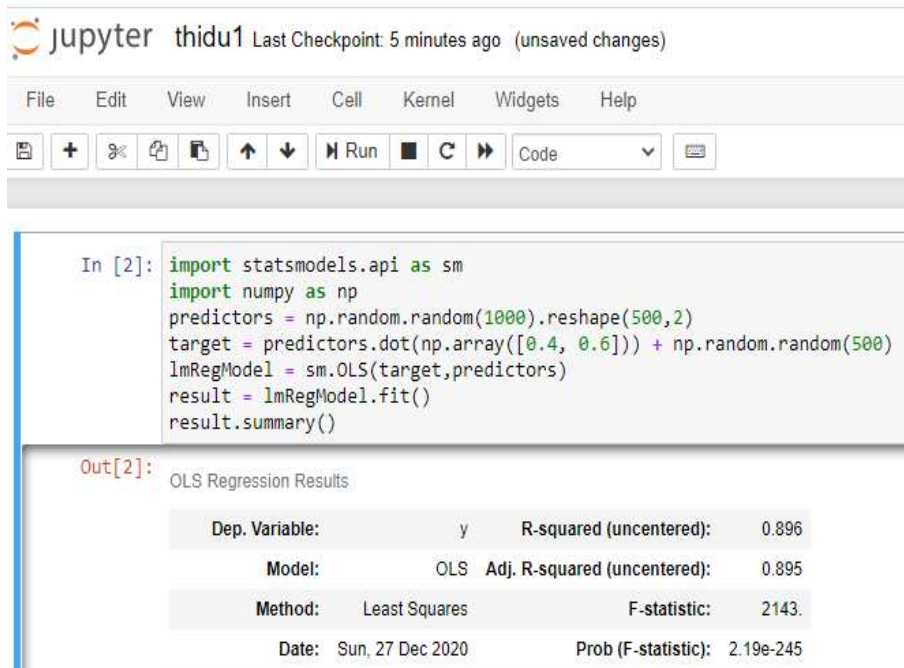


The image shows a Jupyter Notebook interface with the title 'thidu1'. The top bar indicates 'Last Checkpoint: a few seconds ago (unsaved changes)'. The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area contains a code cell with the following Python code:

```
In [ ]: import statsmodels.api as sm
import numpy as np
predictors = np.random.random(1000).reshape(500,2)
target = predictors.dot(np.array([0.4, 0.6])) + np.random.random(500)
lmRegModel = sm.OLS(target,predictors)
result = lmRegModel.fit()
result.summary()
```

**Hình 2.27. Các câu lệnh khai thác mô hình dự đoán tuyến tính**

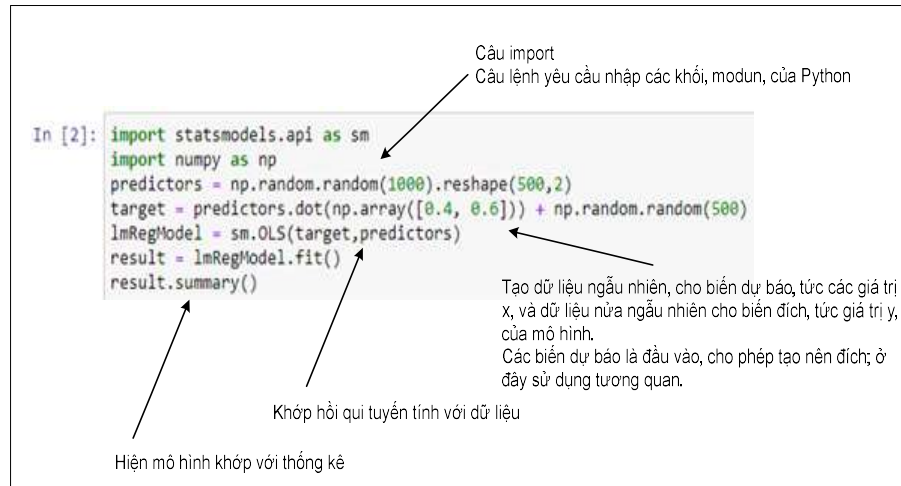
Khi chạy chương trình này, sẽ có kết quả như trong hình 2.28.



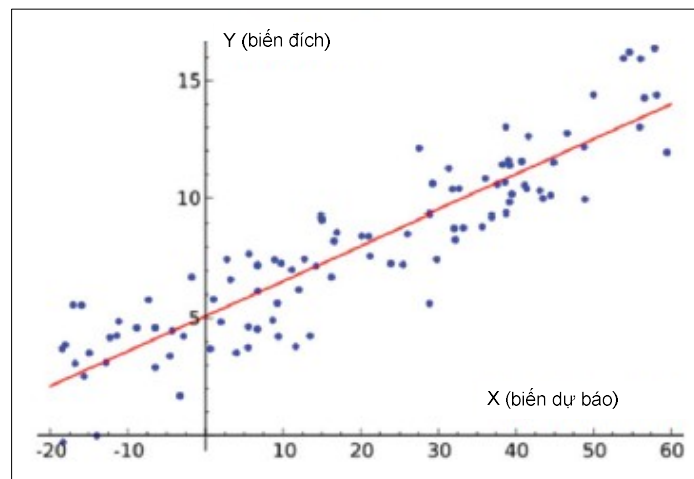
The image shows the same Jupyter Notebook interface as Figure 2.27, but now it displays the output of the code cell. The output is titled 'Out[2]: OLS Regression Results' and contains a table with the following data:

Dep. Variable:	y	R-squared (uncentered):	0.896
Model:	OLS	Adj. R-squared (uncentered):	0.895
Method:	Least Squares	F-statistic:	2143.
Date:	Sun, 27 Dec 2020	Prob (F-statistic):	2.19e-245

**Hình 2.28. Kết quả chạy chương trình**



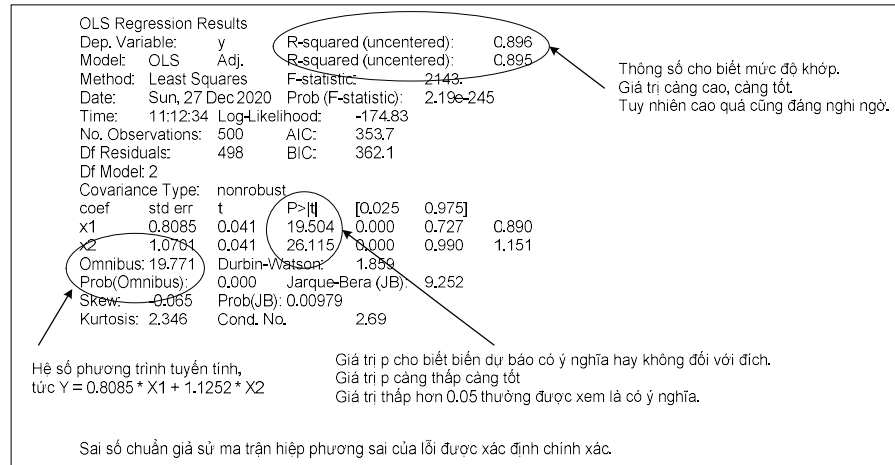
**Hình 2.29. Ý nghĩa của các câu lệnh trong chương trình**



**Hình 2.30. Hồi qui tuyến tính sẽ khớp các dữ liệu với đường thẳng, sao cho sát với các điểm nhất**

Ở đây đã tạo các giá trị dự báo nhằm dự đoán cách các biến mục tiêu hoạt động. Đối với hồi qui tuyến tính, *mối quan hệ tuyến tính* giữa mỗi *biến dự báo x* và *biến mục tiêu y* được giả định. Tuy nhiên, ở đây đã tạo biến mục tiêu dựa trên công cụ dự đoán bằng cách thêm đại lượng ngẫu nhiên. Điều này mang lại một mô hình vừa vặn. Kết quả `results.summary()` xuất ra bảng. Lưu ý, kết quả chính xác phụ thuộc vào các biến ngẫu nhiên nhận được.





**Hình 2.31. Ý nghĩa của giá trị trong kết quả dự báo**

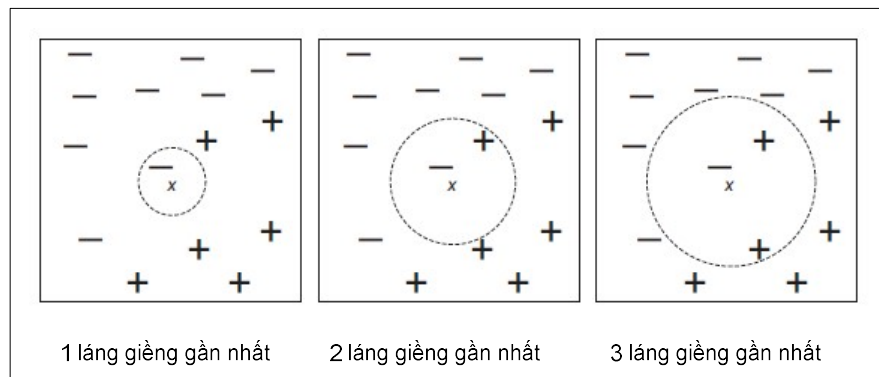
Hãy tạm thời gác lại các kết quả vừa thu được, để tập trung vào những phần quan trọng nhất:

- **Phù hợp của mô hình.** Đối với điều này người ta sử dụng *biên phương R* hoặc *biên phương R điều chỉnh*. Phép đo này là một chỉ báo về lượng biến động trong dữ liệu được mô hình thu thập. Sự khác biệt giữa biên phương R điều chỉnh và biên phương R là tối thiểu ở đây vì giá trị được điều chỉnh là giá trị bình thường cộng thêm *đại lượng phạt*, để xử lý độ phức tạp của mô hình. Một mô hình trở nên phức tạp khi có nhiều biến, hoặc đặc trưng. Người ta không cần một mô hình phức tạp nếu có sẵn một mô hình đơn giản, vì vậy biên phương R điều chỉnh sẽ quá phức tạp. Quy tắc thông thường đối với các mô hình trong doanh nghiệp chấp nhận tỉ lệ trên 0.85 là tốt. Điều quan trọng hơn là ảnh hưởng của các biến dự báo đưa vào;
- **Các biến dự báo có hệ số.** Đối với mô hình tuyến tính, điều này rất dễ hiểu. Trong thí dụ, nếu thêm 1 vào  $x_1$ , nó sẽ thay đổi  $y$  bằng 0.7658. Để nhận thấy việc tìm kiếm một công cụ dự đoán tốt là quan trọng. Thí dụ, nếu xác định rằng một gen nhất định là nguyên nhân gây ung thư, thì đây là kiến thức quan trọng, ngay cả khi bản thân gen đó không xác định liệu một người có bị ung thư hay không. Thí dụ ở đây là phân loại, không phải hồi quy, nhưng cùng ý tưởng: *phát hiện các ảnh hưởng* quan trọng hơn trong các nghiên cứu khoa học hơn là *các mô hình phù hợp* hoàn hảo. Nhưng khi nào biết một gen có tác động đó? Điều này gọi là *ý nghĩa*;
- **Ý nghĩa của dự báo.** Có thể có nhiều hệ số, nhưng đôi khi không chỉ ra được ảnh hưởng ra sao. Chẳng hạn như *giá trị p*. Có thể giải thích dài về sai lầm loại 1 và loại 2 ở đây nhưng các giải thích ngắn gọn sẽ là: nếu giá trị  $p$  thấp hơn 0.05, biến được coi là có ý nghĩa đối với hầu hết mọi người. Trên thực

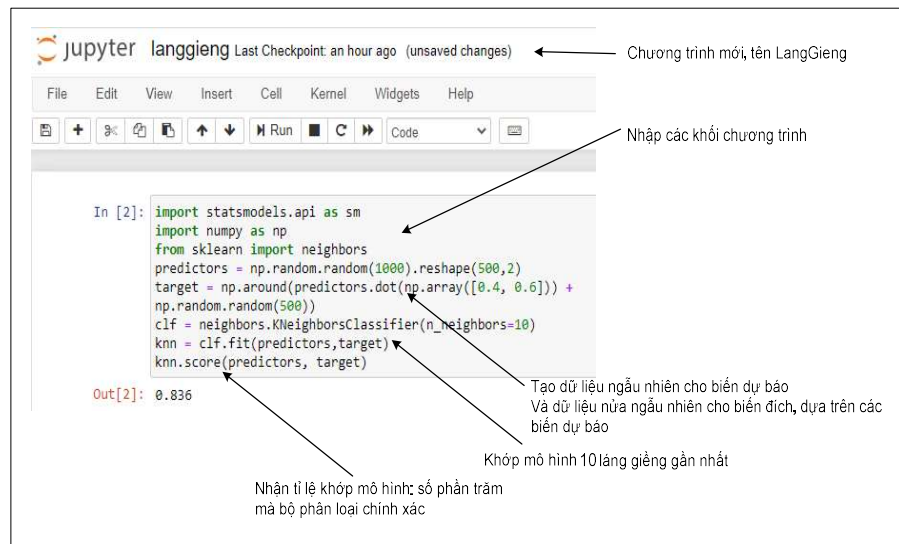
tế, đây là một con số tùy ý. Nó có nghĩa là có 5% khả năng người dự đoán không có bất kỳ ảnh hưởng nào. Liệu có chấp nhận 5% cơ hội sai lầm này không? Còn tùy. Một số người đã đưa ra ngưỡng cực kỳ có ý nghĩa, như  $p < 0.01$ , và ngưỡng có ý nghĩa nhẹ, như  $p < 0.1$ ;

Hồi qui tuyến tính thường được dùng nếu muốn dự đoán một giá trị, nhưng nếu muốn phân loại thứ gì đó? Về các mô hình phân loại, k-láng giềng gần nhất là thông dụng.

Kĩ thuật k-láng giềng gần nhất nhìn vào các điểm được gắn nhãn gần một điểm không được gắn nhãn và dựa trên điều này, đưa ra dự đoán về nhãn sẽ là gì.



**Hình 2.32. Kĩ thuật k - láng giềng gần nhất tìm các điểm gần nhất để dự báo**

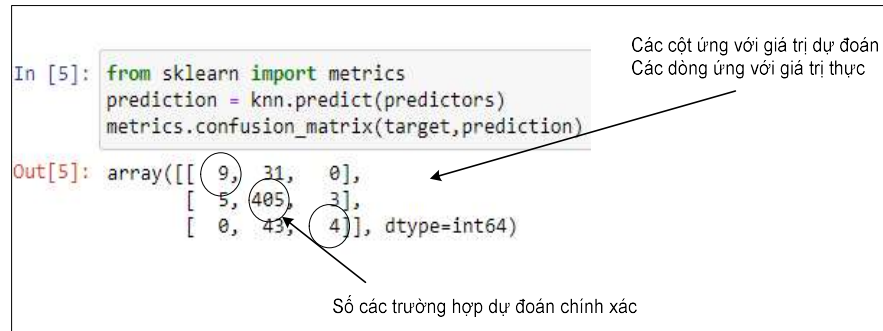


**Hình 2.33. Sử dụng mô hình người láng giềng gần nhất**

Như đã làm, xây dựng dữ liệu tương quan ngẫu nhiên; nhận được 85% trường hợp được phân loại chính xác. Nếu nhìn sâu hơn, cần đánh giá điểm về mô hình. Đừng để `knn.score()` đánh lừa; nó trả về độ chính xác của mô hình, nhưng bằng cách *cho điểm một mô hình*. Người ta thường dùng nó để dự đoán trên dữ

liệu. Dự đoán sẽ là `knn.predict(biến dự đoán)`. Bây giờ có thể dự đoán và so sánh nó với thực tế bằng cách sử dụng ma trận nhầm lẫn. `metrics.confusion_matrix(mục tiêu, dự đoán)`.

Người ta nhận được ma trận  $3 \times 3$  như trong hình 2.34.



**Hình 2.34. Kết quả là ma trận  $3 \times 3$**

*Ma trận nhầm lẫn*<sup>1</sup> cho biết có bao nhiêu trường hợp được phân loại đúng và sai, nhờ so sánh dự đoán với các giá trị thực.

Ma trận nhầm lẫn cho thấy việc dự đoán chính xác  $17 + 405 + 5$  trường hợp, vậy là tốt. Đó không bất ngờ, do những lí do sau:

- Bộ phân loại có ba tùy chọn; đánh dấu sự khác biệt với lần trước `np.around()` sẽ làm tròn dữ liệu thành số nguyên gần nhất của nó. Trong trường hợp này, đó là 0, 1 hoặc 2. Chỉ với 3 tùy chọn, không thể sai nhiều hơn 33% trên 500 lần đoán, ngay cả đối với phân phối ngẫu nhiên thực sự như lật đồng xu;
- Tương quan giữa biến thu được với các yếu tố dự đoán. Bởi vì như cách đã làm, người ta nhận được hầu hết các quan sát là 1. Bằng cách đoán 1 cho mọi trường hợp, người ta đã có một kết quả tương tự;
- Người ta đã so sánh dự đoán với giá trị thực, đúng, nhưng chưa bao giờ dự đoán dựa trên dữ liệu mới. Dự đoán được thực hiện bằng cách sử dụng dữ liệu giống như dữ liệu được sử dụng để xây dựng mô hình. Điều này không cho biết liệu mô hình có hoạt động hay không khi nó gặp phải dữ liệu thực sự mới. Đối với điều này, cần giữ lại một số mẫu để kiểm chứng.

Thành thật mà nói, chỉ một số ít kĩ thuật có triển khai sẵn sàng về công nghiệp bằng Python. Nhưng khá dễ dàng để sử dụng các mô hình có sẵn bằng R trong Python với sự trợ giúp của thư viện RPy. RPy cung cấp giao diện từ Python

<sup>1</sup> Confusion matrix: ma trận nhầm lẫn.

đến R. R là môi trường phần mềm miễn phí, được sử dụng rộng rãi cho tính toán thống kê.

#### *2.2.3.3. Chẩn đoán mô hình và so sánh mô hình*

Sẽ tạo nhiều mô hình từ đó chọn mô hình tốt nhất dựa trên nhiều tiêu chí. Làm việc với một mẫu giữ lại giúp người ta chọn mô hình hoạt động tốt nhất. Mẫu lưu giữ là một phần dữ liệu được lấy ra khỏi quá trình xây dựng mô hình, để có thể sử dụng nó để đánh giá mô hình sau này. Nguyên tắc ở đây rất đơn giản: mô hình phải hoạt động trên dữ liệu chưa gặp. Người ta chỉ sử dụng một phần dữ liệu để ước tính mô hình và phần còn lại, mẫu giữ nguyên, không nằm trong quá trình xây dựng. Sau đó, mô hình được làm việc trên dữ liệu chưa thấy và các phép đo lỗi được tính toán để đánh giá nó. Nhiều phép đo lỗi có sẵn, người ta cho thấy ý tưởng chung về việc so sánh các mô hình. Phép đo sai số được sử dụng trong thí dụ là sai số bình phương trung bình. Sai số bình phương trung bình là một độ đo đơn giản: kiểm tra mọi dự đoán xem nó sai lệch bao xa so với sự thật, bình phương sai số này và cộng sai số của mọi dự đoán. Thí dụ trên so sánh hiệu suất của hai mô hình để dự đoán kích thước đơn hàng từ giá. Mô hình đầu tiên có kích thước là 3 \* giá và mô hình thứ hai là kích thước là 10. Để ước tính các mô hình, người ta sử dụng 80% quan sát được chọn ngẫu nhiên trong số 1000, tức 80%, mà không hiển thị 20% dữ liệu còn lại cho mô hình. Sau khi mô hình được huấn luyện, người ta dự đoán các giá trị cho 20% biến còn lại dựa trên những giá trị mà người ta đã biết giá trị thực và tính toán sai số của mô hình bằng thước đo sai số. Sau đó, chọn mô hình có sai số thấp nhất. Trong thí dụ này, mô hình 1 được chọn vì nó có tổng sai số thấp nhất.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2$$

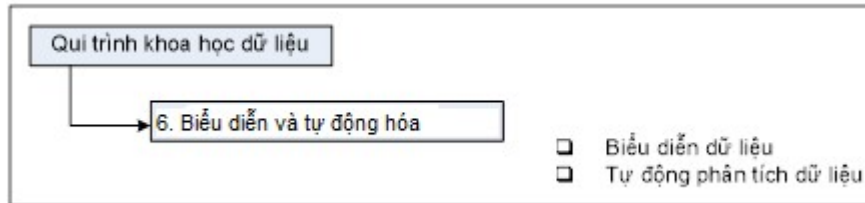
Nhiều mô hình đưa ra các giả định mạnh, chẳng hạn như sự độc lập của các đầu vào và phải xác minh rằng các giả định này thực sự được đáp ứng. Đây được gọi là chẩn đoán mô hình.

#### **2.2.8. Trình bày các phát hiện và xây dựng các ứng dụng**

Sau khi phân tích dữ liệu và xây dựng một mô hình hoạt động tốt, người ta sẵn sàng để trình bày những phát hiện với cộng đồng.

Đôi khi có người quá phấn khích với kết quả đạt được, khiến người ta có nhu cầu lặp lại nó nhiều lần, vì họ coi trọng những dự đoán về mô hình hoặc thông tin chi tiết đã tạo ra. Vì lý do này, cần tự động hóa các mô hình. Điều này không phải lúc nào cũng có nghĩa là phải thực hiện lại tất cả các phân tích mọi lúc. Đôi khi chỉ cần triển khai tính điểm mô hình là đủ; khi khác, có thể tạo một ứng dụng tự động cập nhật báo cáo, bằng tính Excel hoặc bản trình bày PowerPoint. Giai đoạn cuối cùng

của quá trình khoa học dữ liệu là nơi các kĩ năng mềm sẽ hữu ích nhất. Nó cực kỳ quan trọng.



**Hình 2.35. Quá trình biểu diễn dữ liệu**

Phần trên đã nêu quá trình khoa học dữ liệu bao gồm sáu bước:

1. *Đặt mục tiêu nghiên cứu.* Xác định cái gì, tại sao và như thế nào của dự án, theo qui định của dự án;
2. *Truy xuất dữ liệu.* Phát hiện và nhận quyền truy cập vào dữ liệu cần thiết trong dự án. Dữ liệu này được tìm thấy trong công ty hoặc được lấy từ bên thứ ba;
3. *Chuẩn bị dữ liệu.* Kiểm tra và khắc phục lỗi dữ liệu, làm giàu dữ liệu bằng dữ liệu từ các nguồn dữ liệu khác và chuyển đổi nó thành một định dạng phù hợp cho các mô hình trong dự án;
4. *Khám phá dữ liệu.* Đi sâu hơn vào dữ liệu bằng cách sử dụng thống kê mô tả và kĩ thuật hiển thị;
5. *Mô hình hóa dữ liệu.* Sử dụng học máy và kĩ thuật thống kê để đạt được mục tiêu dự án;
6. *Trình bày và tự động hóa.* Trình bày kết quả cho các bên liên quan, cổ đông và công nghiệp hóa quá trình phân tích, để tái sử dụng lặp đi lặp lại và tích hợp với các công cụ khác.

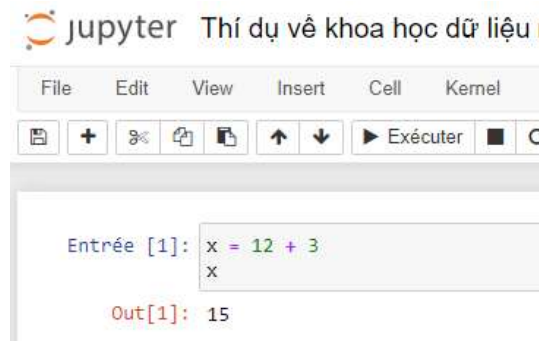
## 2.3. Chuẩn bị dữ liệu với Python

Phần này đề cập việc chuẩn bị dữ liệu, tức một trong sáu bước của quá trình khoa học dữ liệu. Việc chuẩn bị dữ liệu có vai trò quan trọng, nhất là đối với người chưa quen sử dụng công cụ khoa học dữ liệu.

### 2.3.1. Ngôn ngữ Python và môi trường Anaconda, Jupiter

Mục trên trong chương này đã giới thiệu về ngôn ngữ Python. Việc cài đặt người này trên máy cũng đơn giản. Điểm khác biệt của Python so với các ngôn ngữ lập trình khác là ngôn ngữ thông dịch<sup>1</sup>. Python và ngôn ngữ R, ngôn ngữ Julia được cộng đồng chọn như thích hợp với khoa học dữ liệu, mạng vạn vật, dữ liệu lớn...

<sup>1</sup> Phân biệt thông dịch, *interpreter*, với biên dịch, *compiler*.



**Hình 2.36. Giao diện Jupyter**

IPython Notebook hiện được gọi là sổ tay Jupyter. Đây là một môi trường tính toán tương tác, cho phép kết hợp thực thi mã, văn bản đa dạng thức, toán học, đồ thị và đa phương tiện.

### 2.3.2. Thí dụ về chuẩn bị dữ liệu

Công việc chuẩn bị dữ liệu được thực hiện theo nhiều bước, sẽ được giới thiệu kĩ trong mục sau. Tuy nhiên dưới đây là một số bước trên dữ liệu, với minh họa bằng ngôn ngữ Python.

#### 2.3.2.1. Nhập dữ liệu trực tiếp

Thí dụ sử dụng dữ liệu về các đội bóng

- Becamex Bình Dương;
- Hà Nội FC;
- Hải Phòng FC;
- Hoàng Anh Gia Lai;
- Quảng Nam FC;
- Sài Gòn FC;
- Sông Lam Nghệ An;
- Than Quảng Ninh;
- Viettel FC.

Trong mùa giải, người ta có thông tin về tỉ số các trận đấu. Đối với mỗi đội, thông tin này là (i) thắng; (ii) hòa; (iii) thua.

```
In [7]: data = {'Năm': [2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022],
                'Đội': ['HoangAnh Gialai', 'Hanoi FC', 'Haiphong FC', 'SongLam NgheAn',
                        'Viettel FC', 'QuangNam FC', 'Than QuangNinh',
                        'Saigon FC', 'Becamex BinhDuong'],
                'Thắng': [30, 28, 32, 29, 32, 26, 21, 17, 19],
                'Hòa': [6, 7, 4, 5, 4, 7, 8, 10, 8],
                'Thua': [2, 3, 2, 4, 2, 5, 9, 11, 11]}

football = pd.DataFrame(
    data, columns=['Năm', 'Đội', 'Thắng', 'Hòa', 'Thua'])
football
```

```
Out[7]:
```

	Năm	Đội	Thắng	Hòa	Thua
0	2014	HoangAnh Gialai	30	6	2
1	2015	Hanoi FC	28	7	3
2	2016	Haiphong FC	32	4	2
3	2017	SongLam NgheAn	29	5	4
4	2018	Viettel FC	32	4	2
5	2019	QuangNam FC	26	7	5
6	2020	Than QuangNinh	21	8	9
7	2021	Saigon FC	17	10	11
8	2022	Becamex BinhDuong	19	8	11

**Hình 2.37. Nhập dữ liệu trong Python vào thể hiện dữ liệu**

### 2.3.2.2. Nhập dữ liệu từ tệp ngoài

Chẳng hạn có dữ liệu đầu tư cho giáo dục, trên tệp bảng tính Excel. Có thể nhập dữ liệu vào Python bằng cách truy cập đến tệp theo tên.

	A	B	C	D	E
1	Nam	Tỉnh	Chỉ tiêu	DauTu	GhiChu
2	2014	Hà Nội	Phần trăm thu nhập chi cho giáo dục	5.65	
3	2018	Hà Nội	Phần trăm thu nhập chi cho giáo dục	5.58	
4	2020	Hà Nội	Phần trăm thu nhập chi cho giáo dục	6.01	
5	2014	Miền bắc	Phần trăm thu nhập chi cho giáo dục	6.03	
6	2018	Miền bắc	Phần trăm thu nhập chi cho giáo dục	6.42	
7	2020	Miền bắc	Phần trăm thu nhập chi cho giáo dục	7.33	
8	2014	Miền nam	Phần trăm thu nhập chi cho giáo dục	7.21	

**Hình 2.38. Dữ liệu đầu tư cho giáo dục**

```
In [27]: edu = pd.read_csv('E:/thke2/tai_lieu/files/ch02/dulieu.csv', encoding='ISO-8859-1',
                          na_values=':', usecols=['Nam', 'Tỉnh', 'DauTu'])
edu

Out[27]:
```

	Nam	Tỉnh	DauTu
0	2014	Hà Nội	5.65
1	2018	Hà Nội	5.58
2	2020	Hà Nội	6.01
3	2014	Miền bắc	6.03
4	2018	Miền bắc	6.42

**Hình 2.39. Nhập dữ liệu từ tệp ngoài**



Người ta có thể xem dữ liệu đã nhập vào theo đầu danh sách hay cuối danh sách.

In [26]:	edu.head()	In [28]:	edu.tail()																																																
Out[26]:		Out[28]:																																																	
	<table><thead><tr><th></th><th>Nam</th><th>Tinh</th><th>DauTu</th></tr></thead><tbody><tr><td>0</td><td>2014</td><td>Hà Nội</td><td>5.66</td></tr><tr><td>1</td><td>2018</td><td>Hà Nội</td><td>5.58</td></tr><tr><td>2</td><td>2020</td><td>Hà Nội</td><td>6.01</td></tr><tr><td>3</td><td>2014</td><td>Miền bắc</td><td>6.03</td></tr><tr><td>4</td><td>2018</td><td>Miền bắc</td><td>6.42</td></tr></tbody></table>		Nam	Tinh	DauTu	0	2014	Hà Nội	5.66	1	2018	Hà Nội	5.58	2	2020	Hà Nội	6.01	3	2014	Miền bắc	6.03	4	2018	Miền bắc	6.42		<table><thead><tr><th></th><th>Nam</th><th>Tinh</th><th>DauTu</th></tr></thead><tbody><tr><td>25</td><td>2018</td><td>Tây Ninh</td><td>6.43</td></tr><tr><td>26</td><td>2018</td><td>Tây Ninh</td><td>6.34</td></tr><tr><td>27</td><td>2020</td><td>Thanh Hóa</td><td>6.43</td></tr><tr><td>28</td><td>2014</td><td>Thanh Hóa</td><td>5.21</td></tr><tr><td>29</td><td>2018</td><td>Thanh Hóa</td><td>7.20</td></tr></tbody></table>		Nam	Tinh	DauTu	25	2018	Tây Ninh	6.43	26	2018	Tây Ninh	6.34	27	2020	Thanh Hóa	6.43	28	2014	Thanh Hóa	5.21	29	2018	Thanh Hóa	7.20
	Nam	Tinh	DauTu																																																
0	2014	Hà Nội	5.66																																																
1	2018	Hà Nội	5.58																																																
2	2020	Hà Nội	6.01																																																
3	2014	Miền bắc	6.03																																																
4	2018	Miền bắc	6.42																																																
	Nam	Tinh	DauTu																																																
25	2018	Tây Ninh	6.43																																																
26	2018	Tây Ninh	6.34																																																
27	2020	Thanh Hóa	6.43																																																
28	2014	Thanh Hóa	5.21																																																
29	2018	Thanh Hóa	7.20																																																

**Hình 2.40. Xem dữ liệu**

Thông tin về tên các thuộc tính dữ liệu, về chỉ mục dữ liệu như trong hình 2.41.

```
In [29]: edu.columns
Out[29]: Index(['Nam', 'Tinh', 'DauTu'], dtype='object')

In [30]: edu.index
Out[30]: RangeIndex(start=0, stop=30, step=1)
```

**Hình 2.41. Thông tin về dữ liệu**

Dữ liệu được thể hiện ở dạng ma trận. Xem dạng ma trận như trong hình sau.

```
In [31]: edu.values
Out[31]:
Array ([2014, 'Hà Nội', 5.66],
       [2018, 'Hà Nội', 5.58],
       [2020, 'Hà Nội', 6.01],
       [2014, 'Miền bắc', 6.03],
       [2018, 'Miền bắc', 6.42],
       [2020, 'Miền bắc', 7.33],
       [2014, 'Miền nam', 7.21],
       [2018, 'Miền nam', 7.22])
```

**Hình 2.42. Dữ liệu ở dạng ma trận**

### 2.3.2.3. Thống kê

Công cụ thống kê cho phép hiện các giá trị thống kê của tập dữ liệu. Những giá trị chính là (i) số lượng dữ liệu; (ii) giá trị trung bình; (iii) độ lệch; (iv) giá trị cực đại, cực tiểu; (v) giá trị chia phần trăm.

```
In [32]: edu.describe()
```

```
Out[32]:
```

	Nam	DauTu
count	30.000000	28.000000
mean	2017.333333	6.509286
std	2.537081	0.781423
min	2014.000000	5.210000
25%	2014.000000	6.010000
50%	2018.000000	6.430000
75%	2020.000000	7.210000
max	2020.000000	8.030000

**Hình 2.43. Các giá trị thống kê**

```
In [34]: edu['DauTu']
```

```
Out[34]:
```

0	5.65
1	5.58
2	6.01
3	6.03
29	7.20

Name: DauTu, dtype: float64

**Hình 2.44. Hiện dữ liệu theo thuộc tính**

Người ta hiện dữ liệu theo thuộc tính hay theo thứ tự xếp dữ liệu trong danh sách.

```
In [35]: edu[10:14]
```

```
Out[35]:
```

	Nam	Tinh	DauTu
10	2018	Miền trung	6.02
11	2020	Miền trung	6.01
12	2014	Huế	7.02
13	2018	Huế	7.45

**Hình 2.45. Hiện dữ liệu theo thứ tự**

Người ta có thể chọn dữ liệu đồng thời theo thứ tự và theo thuộc tính dữ liệu.

```
In [41]: edu.loc[13:15, ['Nam', 'Tinh']]
```

```
Out[41]:
```

	Nam	Tinh
13 2018		Huế
14 2020		Huế
15 2014		Cần Thơ

**Hình 2.46. Lọc dữ liệu**

Điều kiện tìm kiếm dữ liệu được liệt kê ngay trong biểu thức tên thuộc tính.

```
In [42]: edu[edu['DauTu'] > 6.2].tail()
```

```
Out[42]:
```

	Nam	Tinh	DauTu
24 2014		Tây Ninh	7.02
25 2018		Tây Ninh	6.43
26 2020		Tây Ninh	6.34
27 2014		Thanh Hóa	6.43
29 2020		Thanh Hóa	7.20

**Hình 2.47. Hiện dữ liệu có điều kiện tìm kiếm**

```
In [43]: edu[edu['DauTu'].isnull()].head()
```

```
Out[43]:
```

	Nam	Tinh	DauTu
16 2018		Cần Thơ	NaN
20 2020		Quảng Nam	NaN

**Hình 2.48. Lọc dữ liệu có giá trị thuộc tính rỗng**

```
In [44]: edu.max(axis=0)
```

```
Out[44]:
```

Nam	2020
Tinh	Tây Ninh
DauTu	8.03

dtype: object

**Hình 2.49. Hiện đối tượng đạt giá trị cực đại**

```
In [46]: s = edu['DauTu'] / 100
s.head()
```

```
Out[46]: 0    0.0565
1    0.0558
2    0.0601
3    0.0603
4    0.0642
Name: DauTu, dtype: float64
```

Hình 2.50. Hiện một số giá trị đầu, theo điều kiện

```
In [47]: s = edu['DauTu'].apply(np.sqrt)
s.head()
```

```
Out[47]: 0    2.376973
1    2.362202
2    2.451530
3    2.455606
4    2.533772
Name: DauTu, dtype: float64
```

Hình 2.51. Hiện dữ liệu theo điều kiện phức tạp hơn

```
In [48]: edu['Chuan'] = edu['DauTu'] / edu['DauTu'].max()
edu.tail()
```

```
Out[48]:
```

	Nam	Tinh	DauTu	Chuan
25	2018	Tây Ninh	6.43	0.800747
26	2020	Tây Ninh	6.34	0.789539
27	2014	Thanh Hóa	6.43	0.800747
28	2018	Thanh Hóa	5.21	0.648817
29	2020	Thanh Hóa	7.20	0.896638

Hình 2.52. Hiện một số giá trị cuối danh sách, theo biểu thức giá trị

```
In [49]: edu.drop('Chuan', axis=1, inplace=True)
edu.head()
```

```
Out[49]:
```

	Nam	Tinh	DauTu
0	2014	Hà Nội	5.65
1	2018	Hà Nội	5.58
2	2020	Hà Nội	6.01
3	2014	Miền bắc	6.03
4	2018	Miền bắc	6.42

Hình 2.53. Lọc bớt dữ liệu

#### 2.3.2.4. Xử lý dữ liệu

Một số phép xử lý dữ liệu (i) thêm; (ii) bớt; (iii) sửa dữ liệu được thực hiện trong Python.

```
In [50]: edu = edu.append({'Nam': 2020, 'DauTu': 7.00, 'Tinh': 'Thanh Hóa'}, ignore_index=True)
edu.tail()
```

Out[50]:

	Nam	Tinh	DauTu
26	2020	Tây Ninh	6.34
27	2014	Thanh Hóa	6.43
28	2018	Thanh Hóa	5.21
29	2020	Thanh Hóa	7.20
30	2020	Thanh Hóa	7.00

Hình 2.54. Bỏ sung dữ liệu

```
In [51]: edu.drop(max(edu.index), axis=0, inplace=True)
edu.tail()
```

Out[51]:

	Nam	Tinh	DauTu
25	2018	Tây Ninh	6.43
26	2020	Tây Ninh	6.34
27	2014	Thanh Hóa	6.43
28	2018	Thanh Hóa	5.21
29	2020	Thanh Hóa	7.20

Hình 2.55. Loại bỏ dữ liệu

```
In [52]: eduDrop = edu.drop(edu['DauTu'].isnull(), axis=0)
eduDrop.head()
```

Hình 2.56. Loại bỏ dữ liệu có giá trị rỗng

```
In [53]: eduDrop = edu.dropna(how='any', subset=['DauTu'], axis=0)
eduDrop.head()
```

Out[53]:

	Nam	Tinh	DauTu
0	2014	Hà Nội	5.65
1	2018	Hà Nội	5.58
2	2020	Hà Nội	6.01
3	2014	Miền bắc	6.03
4	2018	Miền bắc	6.42

Hình 2.57. Hiện dữ liệu loại bỏ

```
In [54]: eduFilled = edu.fillna(value={'DauTu': 0})
eduFilled.head()
```

```
Out[54]:
```

	Nam	Tinh	DauTu
0	2014	Hà Nội	5.65
1	2018	Hà Nội	5.58
2	2020	Hà Nội	6.01
3	2014	Miền bắc	6.03
4	2018	Miền bắc	6.42

Hình 2.58. Điền dữ liệu

```
In [55]: edu.sort_values(by='DauTu', ascending=False, inplace=True)
edu.head()
```

```
Out[55]:
```

	Nam	Tinh	DauTu
14	2020	Huế	8.03
8	2020	Miền nam	7.54
17	2020	Cần Thơ	7.54
13	2018	Huế	7.45
5	2020	Miền bắc	7.33

Hình 2.59. Sắp xếp dữ liệu theo giá trị giảm dần

```
In [56]: edu.sort_index(axis=0, ascending=True, inplace=True)
edu.head()
```

```
Out[56]:
```

	Nam	Tinh	DauTu
0	2014	Hà Nội	5.65
1	2018	Hà Nội	5.58
2	2020	Hà Nội	6.01
3	2014	Miền bắc	6.03
4	2018	Miền bắc	6.42

Hình 2.60. Sắp xếp dữ liệu theo giá trị tăng dần

```
In [57]: group = edu[['Tinh', 'DauTu']].groupby('Tinh').mean()
group.head()
```

```
Out[57]:
```

Tinh	DauTu
Đà Nẵng	5.613333
Cần Thơ	7.375000
Huế	7.500000
Hà Nội	5.746667
Miền bắc	6.593333

**Hình 2.61. Nhóm dữ liệu**

```
In [58]: filtered_data = edu[edu['Nam'] > 2015]
pivedu = pd.pivot_table(filtered_data, values='DauTu',
                        index=['Tinh'], columns=['Nam'])
pivedu.head()
```

```
Out[58]:
```

	Nam	2018	2020
Tinh			
Đà Nẵng	5.21	6.20	
Cần Thơ	NaN	7.54	
Huế	7.45	8.03	
Hà Nội	5.58	6.01	
Miền bắc	6.42	7.33	

**Hình 2.62. Tạo bảng xoay**

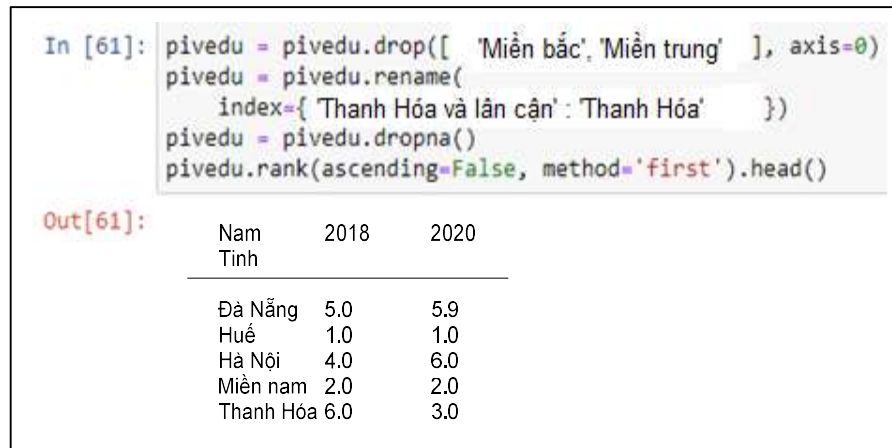
```
In [60]: pivedu.loc[['Thanh Hóa', 'Tây Ninh'], [2018, 2020]]
```

```
Out[60]:
```

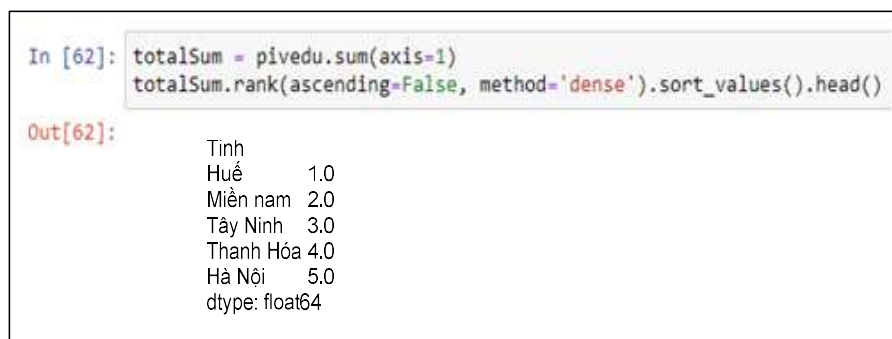
	Nam	2018	2020
Tinh			
Thanh Hóa	5.21	7.20	
Tây Ninh	6.43	6.34	

**Hình 2.63. Một số dữ liệu trong bảng xoay**



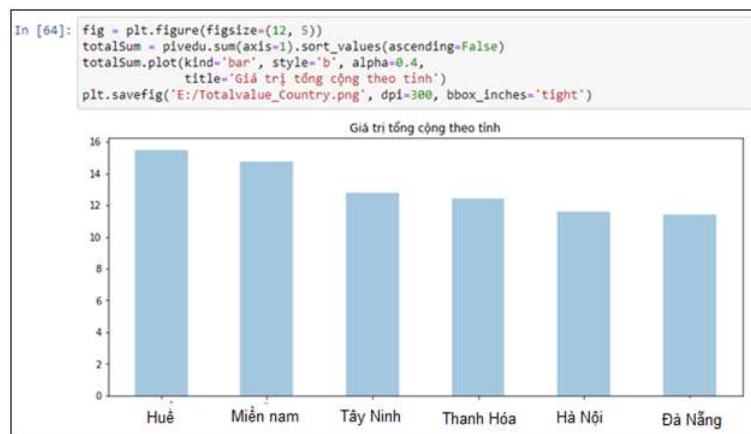


**Hình 2.64. Lọc dữ liệu**

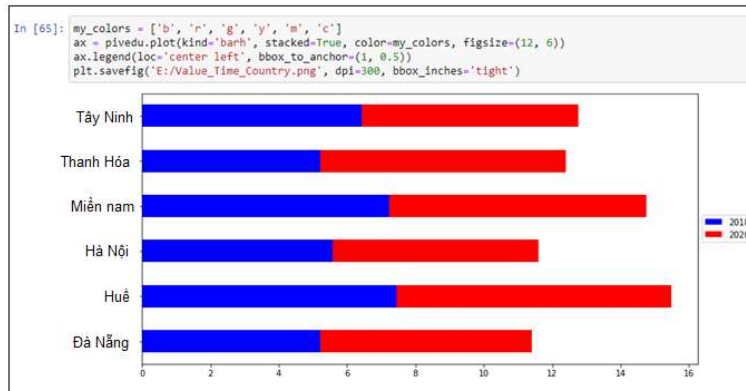


**Hình 2.65. Giá trị tổng cộng theo tỉnh**

### 2.3.3.5. Thể hiện đồ thị



**Hình 2.66. Đồ thị theo giá trị dữ liệu**



**Hình 2.67. Đồ thị thanh ngang**

### 2.3.3. Lấy dữ liệu từ Excel

Sử dụng chức năng liên quan đến dữ liệu trong bảng tính Excel:

```
from pandas import ExcelFile
```

Trích xuất dữ liệu Excel bằng *Pandas.Read\_Excel()*: đã sẵn sàng để tạo đối tượng *DataFrame*. Kết hợp tất cả cùng nhau và đặt đối tượng *DataFrame* thành một biến có tên là DF:

```
DF = pd.read_excel (đường dẫn đến tệp)
```

Cuối cùng, muốn xem *DataFrame*, vì vậy hãy in kết quả. Thêm một câu lệnh in vào cuối đoạn lệnh, sử dụng biến *DataFrame* làm đối số: *print(DF)*

Hãy xem xét kỹ hơn đối tượng *DataFrame*: thoát nhìn, *DataFrame* trông rất giống với bảng Excel thông thường. Điều này khiến *Pandas DataFrame* rất dễ hiểu. Những đầu mục được gắn nhãn ở đầu tập dữ liệu và Python đã điền vào các hàng bằng tất cả thông tin đọc được từ trang tính. Lưu ý cột ngoài cùng bên trái, một chỉ mục bắt đầu từ 0 và đánh số các cột. Theo mặc định, *Pandas* sẽ áp dụng chỉ mục này cho *DataFrame*, có thể hữu ích trong một số trường hợp. Nếu không muốn chỉ mục này được tạo, có thể thêm một đối số bổ sung vào code:

```
DF = pd.read_excel (đường dẫn đến tệp, index=False)
```

## 2.4. Chuẩn bị dữ liệu cho khoa học dữ liệu

Chuẩn bị dữ liệu cho khoa học dữ liệu cũng được xem như chuẩn bị dữ liệu cho việc khai phá dữ liệu, chú trọng vào mục tiêu cần được giải quyết, chứ không chú trọng nhiều đến công nghệ được sử dụng. Nếu không xác định được mục tiêu, tức vấn đề cần giải quyết, sẽ không thể định nghĩa được cách thức trích xuất ra những giá trị từ các hoạt động phân tích dữ liệu. Việc quan trọng không kém là cần xác định được dạng thức của giải pháp. Nếu không có một ý tưởng vững chắc về mục tiêu của mình, sẽ khó có thể xác định được liệu kết quả được tìm thấy và hình thức của nó được đưa ra thực sự là phù hợp hay không. Sau khi (i) xác định được

giải pháp phù hợp; và (ii) thu thập đầy đủ dữ liệu phù hợp, người ta có thể bắt đầu quá trình khoa học dữ liệu.

Khoa học dữ liệu cần xử lý số liệu với mức độ lớn hơn hoặc nhỏ hơn phản ánh hoạt động, sự kiện hoặc đối tượng trong thế giới thực. Trong phạm vi của việc chuẩn bị dữ liệu, nên tập trung hơn vào việc phân tích chính xác hơn việc thể hiện của số liệu, làm thế nào và lý do tại sao chúng được chuyển đổi. Tuy nhiên trước khi thực sự tiến hành việc phân tích việc thao tác dữ liệu, các phần dữ liệu còn thiếu cần được xử lý. Dữ liệu cần được chuẩn bị sao cho dễ truy cập các thông tin lưu trữ, nhờ các công cụ khoa học dữ liệu.

Chuẩn bị dữ liệu không phải là một quá trình thiếu định hướng. Không có bất kỳ công cụ tự động nào mà có thể nhìn vào dữ liệu và thông báo rằng nó phù hợp với dữ liệu. Hy vọng đến một lúc nào đó, các công cụ dựa trên trí tuệ nhân tạo trở nên thông minh hơn, thì may ra việc tự động chuẩn bị dữ liệu mới có khả năng thực hiện được. Hiện nay, việc chuẩn bị dữ liệu vẫn là một nghệ thuật, hơn là kỹ thuật, nhằm thu được bộ dữ liệu tốt. Tuy nhiên, điều này cũng không làm giảm tính quan trọng của các công nghệ và kỹ thuật xử lý số liệu mạnh và hiện đại.

Do việc chuẩn bị dữ liệu không thể được tự động hoàn toàn, người ta thường áp dụng những hiểu biết về đặc trưng của dữ liệu vào dữ liệu đã được chuẩn bị. Những hiểu biết về chức năng và tính khả thi của công cụ chuẩn bị dữ liệu thường quan trọng hơn nhiều cách thức các công cụ đó hoạt động. Các chức năng của bộ công cụ có thể được hiểu như những hộp đen. Cho đến khi các công cụ này thực hiện đáng tin cậy và đúng như dự định, người ta quan tâm nhiều đến hiệu năng, độ tin cậy và giới hạn của các kỹ thuật hơn là sự hiểu biết về cách thức chúng thực hiện.

Mặc dù việc chuẩn bị dữ liệu mang tính nghệ thuật, tuy nhiên để có thể chuẩn bị dữ liệu tốt cần phải có sự hiểu biết tổng thể về các vấn đề, đối tượng được phân tích đồng thời cách thức chúng kết hợp với nhau. Phần này sẽ đưa ra được bức tranh tổng thể về quá trình chuẩn bị dữ liệu. Nó sẽ liên kết các thành phần gồm quá trình phân tích dữ liệu, dữ liệu, bộ dữ liệu và công cụ khai phá dữ liệu thành một thể thống nhất. Tiếp theo sẽ là chi tiết về những gì cần phải làm để chuẩn bị dữ liệu, và làm thế nào để thực hiện điều đó.

#### **2.4.1. Chuẩn bị dữ liệu: đầu vào, đầu ra, mô hình và ra quyết định**

Quá trình chuẩn bị dữ liệu nhằm phân tích đầu vào để thu được đầu ra. Đầu vào bao gồm dữ liệu thô thu được và yêu cầu của người dùng, tức lựa chọn vấn đề, giải pháp có thể, công cụ mô hình, và độ tin cậy... Các kết quả đầu ra chứa dữ liệu và môi trường chuẩn bị thông tin PIE. Các quyết định phải được thực hiện liên quan

đến dữ liệu, các công cụ được sử dụng cho xử lý dữ liệu và các yêu cầu của giải pháp.

Vậy nên dưới đây sẽ trình bày về:

- Đầu vào là gì, kết quả đầu ra là gì, những gì họ làm và lý do;
- Làm thế nào các công cụ mô hình ảnh hưởng đến những gì đã được thực hiện;
- Các giai đoạn chuẩn bị dữ liệu và những gì cần làm ở từng giai đoạn.

Mục đích cơ bản của việc chuẩn bị dữ liệu là xử lý và chuyển đổi *dữ liệu thô* để thông tin ẩn chứa trong dữ liệu có thể được lộ diện hoặc dễ dàng tiếp cận hơn. Cách tốt nhất để thực sự thực hiện thay đổi phụ thuộc vào hai quyết định quan trọng (i) những gì các giải pháp đòi hỏi; và (ii) những gì các công cụ khai thác dữ liệu đòi hỏi. Trong khi những quyết định này ảnh hưởng đến cách dữ liệu được chuẩn bị, đầu vào và đầu ra từ quá trình không bị ảnh hưởng.

*Định nghĩa: Dữ liệu thô<sup>1</sup>, còn được gọi là dữ liệu chính, là dữ liệu, như số, kết quả đọc thiết bị, số liệu..., được thu thập từ một nguồn.*

Bằng cách bỏ qua các chi tiết của quá trình chuẩn bị thực tế ở giai đoạn này, sẽ dễ dàng hiểu lý do cần thiết của mỗi đầu vào và đầu ra tương ứng. Cần cố gắng làm rõ được mối quan hệ giữa tất cả các thành phần, và vai trò của từng thành phần. Với sự sắp xếp các thành phần trong toàn hệ thống chuẩn bị dữ liệu, người ta hiểu dễ dàng về sự cần thiết của từng bước của quá trình chuẩn bị và làm thế nào để các thành phần này thích hợp với tổng thể hệ thống.

Các phần trên đã cho biết quá trình khoa học dữ liệu có sáu bước. Mỗi bước có những yêu cầu khác nhau và được thực hiện độc lập với nhau đối với việc chuẩn bị dữ liệu, bước trước sẽ hoàn thành trước khi bắt đầu bước tiếp theo. Việc thực hiện các bước trên có thể được lặp đi lặp lại đến khi mô hình tối ưu được tìm thấy. Kết quả của mỗi chu kỳ khoa học dữ liệu được xem như xác định lại vấn đề, giải pháp, hoặc ít ra cũng tìm ra dữ liệu khác, tốt hơn để bắt đầu một chu kỳ phân tích mới.

#### **2.4.2. Công cụ mô hình hóa và chuẩn bị dữ liệu**

Thông thường, các công cụ khác nhau thì phù hợp với các công việc khác nhau. Trước khi xây dựng bất kỳ mô hình, hai câu hỏi đầu tiên nên đặt ra (i) cần phải tìm hiểu điều gì?; (ii) đâu là dữ liệu? Quyết định tìm hiểu điều gì dẫn đến hai vấn đề (i) chính xác điều muốn biết?; (ii) muốn biết theo hình thức nào?

---

<sup>1</sup> Raw data: dữ liệu thô.

*Định nghĩa: Mô hình hóa dữ liệu<sup>1</sup> trong kỹ thuật phần mềm là quá trình tạo ra một mô hình dữ liệu cho một hệ thống thông tin bằng cách áp dụng các kỹ thuật chính thức nhất định.*

Rất nhiều công cụ mô hình hiện đang có sẵn, và mỗi công cụ đều có các tính năng, điểm mạnh và điểm yếu khác nhau. Điều này hiển nhiên là đúng trong hiện tại và có lẽ sẽ không có gì thay đổi trong tương lai.

Trong một thời gian dài, khoa học dữ liệu hay khai phá dữ liệu tập trung phát triển các *thuật toán*. Điều này là tự nhiên vì các thuật toán *học máy* khác nhau đã cạnh tranh với nhau trong giai đoạn đầu phát triển của ngành khai phá dữ liệu. Ngày càng có nhiều công ty phần mềm nhận ra rằng người dùng quan tâm nhiều hơn đến *các vấn đề cần giải quyết* hơn là các thuật toán. Việc tập trung vào các vấn đề cần giải quyết có nghĩa là các công cụ mới đang được xây dựng và đóng gói phù hợp các nhu cầu cụ thể thay thế cho các công cụ khai thác dữ liệu có mục đích chung. Thí dụ như các công cụ cụ thể để phân đoạn thị trường trong tiếp thị cơ sở dữ liệu, phát hiện gian lận trong các giao dịch tín dụng, phân tích người dùng rời mạng cho các công ty điện thoại, phân tích và dự đoán thị trường chứng khoán,... Tuy nhiên, những ứng dụng được gọi là *thị trường theo chiều dọc* tập trung vào miền ứng dụng cụ thể cũng có những hạn chế. Khi trở nên có hiệu quả hơn trong các lĩnh vực cụ thể, thông thường bằng cách kết hợp tri thức về miền, ứng dụng đó sẽ khó có thể sử dụng hiệu quả cho các miền ứng dụng khác.

Điều này có nghĩa là người phân tích dữ liệu phải quan tâm tác vụ chuẩn bị dữ liệu trước khi đưa vào mô hình, đặc biệt trong trường hợp dữ liệu được chuẩn bị *tự động* mà không có sự tác động của người dùng.

*Định nghĩa: Phân tích<sup>2</sup> là quá trình chia một chủ đề hoặc nội dung phức tạp thành các phần nhỏ hơn để hiểu rõ hơn về nó.*

Hãy xem xét một hệ thống tự động hóa giao dịch tương lai. Nó có thể dự định để dự đoán chuyển động, xu hướng, và xác suất của lợi nhuận để mở rộng một thị trường tương lai cụ thể. Một số loại mô hình lai hoạt động tốt trong kịch bản như vậy. Nếu bao gồm cả giá thị trường trong quá khứ và hiện tại, chúng được coi là các biến số liên tục và có thể được mô phỏng theo cách tiếp cận dựa trên mạng nơ ron. Hệ thống tổng thể cũng có thể sử dụng đầu vào từ các câu chuyện được phân loại tin tức lấy ra một chuỗi các tin tức. Các tin tức được đọc, phân loại, và xếp hạng theo một số tiêu chí. Dữ liệu phân loại như vậy được mô phỏng tốt hơn bằng cách sử dụng một trong các công cụ trích xuất *luật*. Kết quả của quá trình cần được chuẩn bị trước khi đưa vào một số giai đoạn tiếp theo. Người sử dụng không thấy được tính kỹ thuật cơ bản, nhưng người xây dựng hệ thống sẽ phải thực hiện nhiều

---

<sup>1</sup> Data modeling: mô hình hóa dữ liệu.

<sup>2</sup> Analysis: phân tích.

lựa chọn, bao gồm cả những kĩ thuật chuẩn bị dữ liệu tối ưu đáp ứng từng mục tiêu. Dữ liệu *phạm trù* và dữ liệu *số* có thể tốt và thường, đòi hỏi các kĩ thuật chuẩn bị khác nhau.

*Định nghĩa: Biến phân loại<sup>1</sup>, hay biến phạm trù, là một biến có thể nhận một trong số giới hạn và thường cố định, số lượng giá trị có thể có, gán mỗi cá nhân hoặc đơn vị quan sát khác cho một nhóm cụ thể hoặc danh mục trên cơ sở một số thuộc tính định tính.*

Ở giai đoạn thiết kế dự án, hoặc khi trực tiếp sử dụng các công cụ/thư viện chung, điều quan trọng là phải nhận thức được nhu cầu, điểm mạnh và điểm yếu của mỗi công cụ được sử dụng. Mỗi công cụ có một đầu ra hơi khác nhau. Rất khó trong việc đưa ra các luật dễ hiểu từ mạng nơ ron so với tập luật được trích xuất ra từ công cụ phân tích tập luật khác. Hầu như chắc chắn có thể chuyển đổi kết quả đầu ra của công cụ này sang đầu ra của công cụ khác. Tuy nhiên tốt nhất nên sử dụng một công cụ cung cấp loại đầu ra mà người dùng mong muốn.

#### *2.4.2.1. Cách mô hình hóa công cụ chuẩn bị dữ liệu*

Có nhiều loại công cụ mô hình hóa. Mỗi công cụ đều có thể mạnh và những điểm yếu. Điều quan trọng là phải hiểu những đặc điểm cụ thể nào của mỗi công cụ ảnh hưởng đến việc chuẩn bị dữ liệu.

Một yếu tố chính mà các công cụ khoa học dữ liệu ảnh hưởng đến việc chuẩn bị dữ liệu là cách thức công cụ phân biệt giá trị kiểu số/ phân loại. Thứ hai là khả năng thích ứng với các giá trị còn thiếu. Để hiểu tại sao những phân biệt này là quan trọng, nên tìm hiểu cách thức mô hình phân tích dữ liệu cho ra kết quả.

Cách thức mà các công cụ mô hình xử lý các mối quan hệ giữa các biến là phân vùng dữ liệu sao cho dữ liệu trong các phân vùng đặc biệt có thể tương quan với kết quả đầu ra. Do dữ liệu có thể là các giá trị rời rạc hoặc liên tục nên các công cụ mô hình cũng có loại phù hợp với dữ liệu rời rạc hoặc liên tục.

Vậy có thể xem xét và đánh giá một số công cụ, để giúp hiểu được cách thức làm việc của từng công cụ, nhằm thu được kết quả mong muốn.

#### *2.4.2.2 Cây quyết định*

Cây quyết định sử dụng một phương pháp liên kết logic để xác định các vùng của không gian trạng thái. Những liên kết logic này có thể được biểu diễn dưới dạng luật IF... THEN... Nói chung, một cây quyết định xem xét các biến riêng lẻ, từng biến một. Nó bắt đầu bằng cách tìm biến phân chia tốt nhất không gian trạng thái và tạo ra một luật của để xác định sự phân chia. Thuật toán cây quyết định tìm

---

<sup>1</sup> Category variable: biến phạm trù, biến phân loại.

cho mỗi tập hợp con của các thể hiện một luật phân tách. Điều này tiếp tục cho đến khi kích hoạt hết các tiêu chí

#### *2.4.2.3. Danh sách quyết định*

Danh sách quyết định cũng tạo ra các luật IF... THEN... với thể hiện đồ họa như cây quyết định. Tuy nhiên, các cây quyết định xem xét việc phân chia lực lượng ra phần trái, phần phải và chia tách chúng. Danh sách quyết định thường tìm một luật để mô tả rõ một số phần nhỏ lực lượng sau đó được loại bỏ khỏi xem xét thêm. Tại thời điểm đó, nó tìm kiếm một luật khác cho một số phần còn lại.

#### *2.4.2.4. Mạng nơ ron*

Mạng nơ ron<sup>1</sup>, hay mạng thần kinh nhân tạo, cho phép không gian trạng thái được chia thành các phân đoạn với các vết chia không song song với các trục. Điều này được thực hiện thông qua việc mạng nơ ron tìm hiểu một loạt các trọng số của tại mỗi nút. Kết quả của việc này là huấn luyện mạng tạo ra độ dốc hoặc đường dốc để phân đoạn không gian trạng thái. Trong thực tế, các dạng mạng thần kinh phức tạp hơn có thể học cách điều chỉnh các đường cong thông qua không gian trạng thái. Điều này cho phép linh hoạt khi tìm cách xây dựng phân khúc tối ưu.

#### *2.4.2.5. Chương trình tiến hóa*

Trong thực tế, sử dụng một kĩ thuật gọi là *lập trình tiến hóa*<sup>2</sup> có thể thực hiện một loại hồi qui được gọi là *hồi qui theo kí hiệu*<sup>3</sup>. Nó không có nhiều điểm chung với quá trình tìm phương trình hồi qui được sử dụng trong phân tích thống kê, nhưng cho phép phát hiện ra các mối quan hệ đặc biệt khó.

#### *2.4.2.6. Mô hình hóa dữ liệu nhờ công cụ*

Có nhiều kĩ thuật hơn các công cụ được liệt kê ở đây; tuy nhiên, đó là những đại diện của các kĩ thuật được sử dụng trong các công cụ khai thác dữ liệu và khoa học dữ liệu. Tất cả đều mở rộng những ý tưởng cơ bản theo cách mà nhà cung cấp cảm thấy tăng cường hiệu năng của thuật toán cơ bản. Những công cụ này cho phép chuẩn bị dữ liệu theo những cách khác nhau.

Được đánh giá ở mức cao, các công cụ mô hình hoá dữ liệu riêng biệt bằng cách sử dụng một trong hai cách tiếp cận:

- Cách thứ nhất mà công cụ sử dụng là tạo ra một số vết chia trong bộ dữ liệu, tách biệt tổng số dữ liệu được chia thành nhiều phần. Việc chia này tiếp tục cho đến khi đáp ứng được một số tiêu chí dừng;

---

<sup>1</sup> Neural network: mạng nơ ron, mạng thần kinh nhân tạo.

<sup>2</sup> Evolution programming: lập trình tiến hóa.

<sup>3</sup> Symbolic regression: hồi qui theo kí hiệu.



- Cách thứ hai là tạo nên sự phù hợp với một bề mặt linh hoạt, hoặc ít nhất là một phần mở rộng chiều cao của một đa tạp, giữa các điểm dữ liệu để tách chúng.

Điều quan trọng cần lưu ý là trong thực tế không thể chỉ với các thông tin có trong bộ dữ liệu mà tách tất cả các điểm một cách hoàn hảo. Thông thường, sự tách biệt hoàn hảo là không thực sự như ý. Vì tín hiệu ồn, vị trí của nhiều điểm có thể không thực sự thể hiện ý nghĩa thực sự, nên ảnh hưởng xấu đến kết quả phân tách. Để tìm một sự phù hợp hoàn hảo sẽ cần biết về tín hiệu ồn này.

Sự khác biệt chính giữa các công cụ là (i) các công cụ rời rạc, chia dữ liệu đặt vào các vùng rời rạc; (ii) nhạy cảm với sự khác biệt về xếp hạng hoặc thứ tự của các giá trị trong các biến số. Sự khác biệt định lượng không có ảnh hưởng. Những công cụ như vậy tạo điều kiện thuận lợi hay không tùy vào hoàn cảnh. Một danh sách xếp hạng của khoảng cách chung giữa các thành phố mang đủ thông tin để phục hồi bố cục địa lí rất chính xác. Vì vậy, sự khác biệt về xếp hạng mang nội dung thông tin cao.

- Các công cụ rời rạc không phải là đặc biệt gặp rắc rối bởi các giá trị ngoại lai vì nó coi trọng xếp hạng các vị trí. Một mặt do nằm ở vị trí xếp hạng đã xác định bất kể giá trị của nó; mặt khác, các công cụ rời rạc không nhìn thấy định lượng sự khác biệt giữa các giá trị, không thể kiểm tra cấu trúc tinh vi nhúng vào đó. Nếu đó là nội dung thông tin cao trong sự khác biệt định lượng giữa các giá trị, một công cụ có thể mô hình liên tục là cần thiết;
- Các công cụ liên tục có thể trích xuất cả định lượng và định tính, hoặc xếp hạng, nhưng rất nhạy cảm với các loại biến dạng khác nhau trong bộ dữ liệu, chẳng hạn như các ngoại lệ. Việc lựa chọn công cụ phụ thuộc rất nhiều vào tính chất của dữ liệu cùng với các yêu cầu của bài toán.

Trong thực tế, các nhà tạo nên công cụ khoa học dữ liệu đã có những sửa đổi khác nhau để đáp ứng các nhu cầu riêng đối với mỗi loại công cụ.

#### *2.4.2.7. Dự đoán và các qui luật*

Lựa chọn công cụ tác động quan trọng đến những kĩ thuật được áp dụng cho dữ liệu chưa được chuẩn bị. Tất cả các kĩ thuật được mô tả ở đây tạo nên ra một trong hai hình thức (i) dự báo; hoặc (ii) các qui luật. Các công cụ mô hình hoá dữ liệu kết thúc bằng việc học máy như (i) một số dự đoán; (ii) phân loại nhờ dự đoán; (iii) bộ luật có thể được sử dụng để tách dữ liệu một cách hữu ích.

Một công cụ thông dụng là *mô hình nhị phân*, có/không có đầu ra. Công cụ thích hợp nhất là một số loại phân loại cho phép phân loại hồ sơ theo các yêu cầu

nhị phân. Chuẩn bị dữ liệu để phân loại nhị phân có thể có lợi từ các kĩ thuật như *xếp ngăn*<sup>1</sup>, làm tăng khả năng của nhiều công cụ tách các dữ liệu. Xếp ngăn là một kĩ thuật phân chia các phạm vi giá trị thành các ngăn, hoặc *thùng*, với mục đích làm giảm sự biến đổi, loại bỏ một số cấu trúc, trong một bộ dữ liệu. Thí dụ, thẻ thông tin phản hồi của khách hàng thường yêu cầu hộ gia đình thu nhập bằng cách sử dụng các phạm vi *từ-tới* trong đó thu nhập hộ gia đình giảm. Những phạm trù này được xếp ngăn, cho phép xếp hạng thu nhập.

Các phương pháp lập mô hình liên tục và nhị phân có thể được sử dụng, thay vì chỉ *dự đoán*. Khi xây dựng mô hình để hiểu những gì *điều hướng* các hiệu ứng nhất định trong bộ dữ liệu, các mô hình thường được sử dụng để trả lời các câu hỏi về những đặc tính nào quan trọng trong các khu vực đặc biệt của không gian trạng thái. Các kĩ thuật lập mô hình như vậy được sử dụng để trả lời các câu hỏi như: các yếu tố bên dưới liên quan đến gian lận giao dịch tại các văn phòng chi nhánh? Vì các yếu tố ảnh hưởng có thể từ khu vực này sang khu vực khác trong không gian trạng thái, điều quan trọng là phải sử dụng các kĩ thuật chuẩn bị để giữ lại phần lớn cấu trúc tốt; nghĩa là, các biến động chi tiết trong dữ liệu đặt trong các biến thể của các biến.

*Tìm kiếm các yếu tố ảnh hưởng* là một hình thức mô hình hóa theo *suy luận*. Kiểm tra những gì phổ biến đối với các luật là một cách để khám phá các chủ đề phổ biến hiện có trong tình huống. Có thể xuất hiện lí do hành động đặc biệt quan trọng trong một số trường hợp, chẳng hạn như phê duyệt tín dụng hoặc từ chối, nếu có yêu cầu pháp lí để giải thích. Thế hệ các luật như vậy cũng có thể được thể hiện, như là các câu lệnh SQL, nếu cần để trích xuất các phần của bộ dữ liệu. Điều quan trọng ở đây là kết quả đầu ra chịu tác động như thế nào từ việc chuẩn bị các dữ liệu đầu vào, chứ không phải là sử dụng các giải pháp sẽ được đưa ra.

Định nghĩa: Suy luận<sup>2</sup> là các bước lập luận, chuyển từ tiền đề đến hệ quả logic, là lập luận trong hệ chuyên gia.

#### 2.4.2.8. *Lựa chọn các kĩ thuật*

Do nhu cầu lựa chọn các công cụ mô hình hóa khi chuẩn bị dữ liệu, cần quan tâm đến đặc tính của công cụ. Một công cụ có thể sẽ tốt hơn nếu cung cấp cho chúng (i) dữ liệu số; hay (ii) dữ liệu phạm trù. Vì các công cụ thực tế đã sửa đổi các thuật toán thuần túy, nên không thể mong có công cụ tổng quát. Mỗi công cụ phải được đánh giá riêng biệt. Thực tế có lẽ tốt nhất là (i) thử một số kĩ thuật chuẩn bị; (ii) chuẩn bị dữ liệu liên tục; (iii) sử dụng nhiều tùy chọn xếp ngăn để tạo phân loại dữ

---

<sup>1</sup> Binning: xếp ngăn, thùng.

<sup>2</sup> Inference: suy luận.

liệu. Tuy nhiên, nên sử dụng một công cụ khai thác dữ liệu mà đầu ra phù hợp với nhu cầu của giải pháp.

Trong yêu cầu về dự đoán, công cụ cần phải có khả năng đáp ứng và cần đến phân loại, kiểm tra và chuẩn bị dữ liệu. Các kĩ thuật chuẩn bị dữ liệu cho phép dữ liệu được thao tác khi cần thiết, vì vậy người làm khoa học dữ liệu có thể tập trung sự chú ý về quyết định về (i) kĩ thuật; (ii) công cụ thích hợp để sử dụng dữ liệu trong một tình huống cụ thể.

#### *2.4.2.9. Vấn đề thiếu dữ liệu và các công cụ mô hình hóa*

Thiếu các giá trị tạo nên vấn đề rất quan trọng trong việc chuẩn bị dữ liệu. Bất cứ khi nào có giá trị còn thiếu, điều quan trọng là phải xử lí. Có một số phương pháp để xác định một giá trị thay thế thích hợp, nhưng không có trường hợp nào các giá trị bị thiếu sẽ bị bỏ qua hoặc bỏ đi. Một số công cụ, đặc biệt là những người xử lí các giá trị phân loại tốt, chẳng hạn như cây quyết định, được cho là đã xử lí các giá trị còn thiếu. Một số thực sự có thể; một số khác không thể. Một số mô hình hóa với công cụ rời rạc có thể thực sự dễ bỏ qua các giá trị còn thiếu, trong khi các công cụ khác xem một giá trị còn thiếu chỉ như một giá trị phân loại; điều này thực sự không phải là cách tiếp cận thỏa đáng. Các công cụ khác, chẳng hạn như mạng nơ ron, yêu cầu mỗi đầu vào phải được cung cấp một giá trị số và bất kỳ bản ghi có một giá trị bị thiếu phải được hoàn toàn bỏ qua hoặc sử dụng giá trị mặc định đối với giá trị thiếu.

Sẽ có vấn đề với cách tiếp cận thay thế mặc định được thực hiện; thường là những vấn đề lớn. Ít nhất, không được xử lí, ngoại trừ giải pháp mặc định, khi thiếu các giá trị gây ra biến dạng đáng kể đối với vài bộ dữ liệu. Không phải tất cả các giá trị thiếu; có thể giả định dùng một giá trị để đại diện cho các trường hợp thiếu dữ liệu. Tuy nhiên, đó là những gì một cây quyết định làm nếu nó chỉ định các giá trị còn thiếu cho một loại riêng biệt; giả định rằng tất cả chúng đều có cùng giá trị đo. Sự biến dạng tương tự xảy ra nếu một số giá trị số mặc định được gán. Rõ ràng, cần tìm một giải pháp tốt. Một số lựa chọn có sẵn, và những ưu, nhược điểm của mỗi phương pháp nên được đề cập. Đây là một trong những vấn đề phải được giải quyết một cách có hiệu quả để xây dựng được mô hình.

#### **2.4.3. Các giai đoạn chuẩn bị dữ liệu**

Việc chuẩn bị dữ liệu tốt có ý nghĩa đối với kết quả của quá trình khoa học dữ liệu, đặc biệt đối với việc xây dựng được mô hình về dữ liệu. Chuẩn bị dữ liệu bao gồm hai nhóm các hoạt động chuẩn bị:

- Các hoạt động chuẩn bị không liên quan đến quá trình, hoặc hoạt động dẫn đến quyết định về phương pháp tiếp cận mà người phân tích dữ liệu

thực hiện. Có rất nhiều hoạt động và quyết định được thực hiện trong giai đoạn này được mô tả là *chuẩn bị cơ bản*;

- Các hoạt động chuẩn bị tự động. Sau đây sẽ mô tả chi tiết các kĩ thuật được sử dụng trong giai đoạn này.

Các giai đoạn chuẩn bị dữ liệu được chia ra:

1. Truy cập dữ liệu;
2. Kiểm tra dữ liệu;
3. Tăng cường và làm giàu dữ liệu;
4. Tìm kiếm mẫu thiên lệch;
5. Xác định cấu trúc dữ liệu;
6. Xây dựng PIE;
7. Khảo sát dữ liệu;
8. Lập mô hình dữ liệu.

Định nghĩa: Việc chuẩn bị<sup>1</sup> là làm sẵn sàng sử dụng đối với các thành phần sẽ được thực hiện.

#### *2.4.3.1. Giai đoạn 1: Truy cập dữ liệu*

Điểm khởi đầu cho bất kỳ dự án chuẩn bị dữ liệu là xác định vị trí dữ liệu. Điều này đôi khi không dễ như người ta nghĩ. Có một số vấn đề cản trở truy cập vào dữ liệu được chỉ định, từ hợp pháp đến kết nối. Một số vấn đề thường gặp phải được xem xét sau, nhưng đánh giá toàn diện về tất cả các vấn đề gần như không thể, đơn giản vì mỗi dự án đều cung cấp các tình huống duy nhất. Tuy nhiên, việc định vị và đảm bảo nguồn cung cấp dữ liệu và đảm bảo truy cập đầy đủ không chỉ là bước đầu tiên, nó thực sự cần thiết.

Có một phần của vấn đề bị lộ do người có quyền truy cập vào *kho dữ liệu*. Một thực tế là kho dữ liệu đang trở thành kho lưu trữ được lựa chọn, *kho dữ liệu chuyên đề*. Càng ngày nó càng là một kho được khai thác. Tuy nhiên, một kho không có nghĩa là thiết yếu để khai thác dữ liệu. Trong thực tế, một kho có thể gây bất lợi cho nỗ lực khai thác, tùy thuộc vào cách dữ liệu được tải. Kho cũng có những nhược điểm khác, một điều quan trọng là chúng thường được tạo ra với một cấu trúc cụ thể để phản ánh một số quan điểm cụ thể về doanh nghiệp. Cấu trúc áp đặt này có thể ảnh hưởng thiên lệch đối với kết quả mô hình hóa.

*Định nghĩa: Truy cập dữ liệu<sup>2</sup> là một thuật ngữ chung đề cập đến một quá trình có cả ý nghĩa cụ thể về công nghệ thông tin và các ý nghĩa khác liên quan đến quyền truy cập theo nghĩa rộng hơn về mặt pháp lý và chính sách.*

---

<sup>1</sup> Preparation: chuẩn bị.

<sup>2</sup> Data access: truy cập dữ liệu. Truy cập mang ý (i) truy nhập; (ii) truy xuất; (iii) tham chiếu.

#### 2.4.3.2. Giai đoạn 2: Kiểm tra dữ liệu

Giả sử có sẵn dữ liệu phù hợp, tập hợp các vấn đề cơ bản đầu tiên phải được giải quyết:

- Nguồn cung cấp;
- Số lượng dữ liệu;
- Chất lượng của dữ liệu.

Xây dựng các mô hình mạnh đòi hỏi dữ liệu đủ về số lượng và chất lượng đủ cao để tạo ra mô hình cần thiết. Kiểm toán dữ liệu là một phương pháp xác định trạng thái của bộ dữ liệu và ước tính mức độ phù hợp của nó cho việc xây dựng mô hình. Thực tế là kiểm toán dữ liệu không đảm bảo rằng mô hình sẽ có thể được xây dựng, nhưng ít nhất đảm bảo rằng các yêu cầu tối thiểu đã được đáp ứng.

*Định nghĩa: kiểm tra dữ liệu<sup>1</sup> là việc tra xét kỹ lưỡng về sự đúng sai.*

Kiểm toán yêu cầu kiểm tra các mẫu dữ liệu nhỏ và đánh giá các trường cho nhiều tính năng, chẳng hạn như số trường, nội dung của từng trường, nguồn của từng trường, giá trị tối đa và tối thiểu, số giá trị rời rạc và nhiều số liệu cơ bản khác.

*Định nghĩa: Kiểm toán dữ liệu<sup>2</sup> là quá trình thực hiện kiểm tra dữ liệu để đánh giá xem dữ liệu của công ty có phù hợp với mục đích nhất định hay không.*

Khi dữ liệu đã được đánh giá về số lượng và chất lượng, một câu hỏi chính được đặt ra là (i) lí do chính đáng để cho rằng dữ liệu đã thu thập có khả năng cung cấp giải pháp cần thiết cho vấn đề; (ii) đã đủ dữ liệu? Đây là vấn đề quan trọng để loại bỏ sự kỳ vọng về kết quả ngẫu nhiên. Không nên hy vọng rằng: bộ dữ liệu có sẵn sẽ thực sự giữ một cái gì đó có giá trị để dẫn đến một mô hình thỏa đáng. Một phần quan trọng của kiểm toán, một phần phi kĩ thuật, là xác định tính khả thi thực sự của việc cung cấp giá trị với các tài nguyên có sẵn. Trên thực tế, cần chỉ ra những lí do đảm bảo dữ liệu thực tế có thể đáp ứng các thách thức của thực tế.

#### 2.4.3.3. Giai đoạn 3: Tăng cường và làm giàu dữ liệu

Một khi đã thực hiện việc kiểm toán đối với dữ liệu, người ta có thể tin tưởng vững chắc về tính đầy đủ của dữ liệu. Nếu kiểm toán tiết lộ rằng dữ liệu không thực sự trợ giúp cho những hy vọng được tạo ra trên đó, có thể bổ sung dữ liệu theo nhiều cách khác nhau. *Thêm dữ liệu* là một cách phổ biến để tăng nội dung thông tin.

Có một số cách mà dữ liệu hiện tại có thể được thao tác để mở rộng tính hữu dụng của nó. Thao tác như vậy cho phép tính tỉ lệ giá/thu nhập, kí hiệu P/E, để mô hình hóa giá trị của giá cổ phiếu. Nhà đầu tư xem tỉ lệ này có giá trị dự đoán. Giá

---

<sup>1</sup> Test: kiểm tra, thử nghiệm, kiểm định.

<sup>2</sup> Data audit: kiểm toán dữ liệu.

và thu nhập có sẵn trong dữ liệu nguồn, nên việc cung cấp thông tin về tỉ lệ P/E sẽ giúp ích để ra quyết định (i) tỉ lệ P/E thể hiện cái nhìn sâu sắc về miền ứng dụng về những gì là quan trọng; cho phép có thêm thông tin đối với công cụ mô hình hóa đầu vào; (ii) trình bày thông tin được tính toán, giúp công cụ lập mô hình không phải học phân loại.

Công cụ mô hình hóa có thể và học nhiều lần về các mối quan hệ. Chúng có thể học các mối quan hệ phức tạp. Tuy nhiên, cần có thời gian và tài nguyên hệ thống để khám phá bất kỳ mối quan hệ nào. Thêm đủ kiến thức liên quan đến lĩnh vực ứng dụng và trợ giúp học tập về các tính năng quan trọng có thể tăng hiệu suất và giảm đáng kể thời gian phát triển. Trong một số trường hợp, nó giúp tạo ra các mô hình hữu ích.

#### *2.4.3.4. Giai đoạn 4: Tìm kiếm mẫu thiên lệch*

Sai lệch khi lấy mẫu tạo nên một số vấn đề đặc biệt phức tạp. Có một số phương pháp tự động giúp phát hiện sai lệch lấy mẫu, nhưng không có phương pháp tự động nào có thể khớp với suy nghĩ hợp lí.

*Định nghĩa: Thiên lệch<sup>1</sup> là việc ưu tiên không hợp lí cho một giá trị.*

Có nhiều phương pháp lấy mẫu, và lấy mẫu luôn cần thiết. *Lấy mẫu* là quá trình lấy một phần nhỏ của một bộ dữ liệu lớn hơn theo cách mà phần nhỏ phản ánh chính xác mối quan hệ trong bộ dữ liệu lớn. Vấn đề là các mối quan hệ thực sự tồn tại trong tập hợp dữ liệu đầy đủ nhất có thể, được gọi là dân số, có thể vì nhiều lí do, là không thể biết. Điều đó có nghĩa là không thể thực sự kiểm tra xem mẫu có phải là đại diện của dân số trong thực tế. Điều quan trọng là phải nỗ lực để chắc chắn rằng dữ liệu thu được là đại diện cho tình trạng thực sự nhất có thể.

*Định nghĩa: Lấy mẫu<sup>2</sup> là việc lựa chọn một tập hợp con, tức một mẫu thống kê, của các cá thể từ trong một quần thể thống kê để ước tính các đặc điểm của toàn bộ quần thể.*

Trong khi lấy mẫu được trình bày trong nhiều tài liệu về thống kê, các người làm khoa học dữ liệu phải đối mặt với các vấn đề chưa được giải quyết trong các tài liệu ấy. Người ta thường cho rằng *nhà phân tích*, tức nhà thống kê/người lập mô hình, có một số quyền kiểm soát đối với *cách* dữ liệu được tạo và thu thập. Nếu không phải là nhà phân tích, ít nhất người tạo hoặc người thu thập dữ liệu có thể được coi là đã thực hiện kiểm soát phù hợp để tránh lấy *mẫu sai lệch*. Tuy nhiên, các công cụ khoa học dữ liệu đôi khi phải đối mặt với các bộ dữ liệu gần như chắc chắn được thu thập cho các mục đích không xác định, bởi các quá trình là *không chắc chắn*, mà lại dự kiến sẽ trợ giúp đưa ra câu trả lời cho các câu hỏi chưa

---

<sup>1</sup> Bias: thiên lệch.

<sup>2</sup> Sampling: lấy mẫu. Phân biệt nó với Pattern.

được biết đến vào thời điểm đó. Với nguồn gốc của dữ liệu chưa biết, rất khó để đánh giá những nhược điểm trong dữ liệu và nếu xử lý các nhược điểm sẽ tạo ra các mô hình sai lầm và không thể áp dụng.

#### 2.4.3.5. Giai đoạn 5: Xác định cấu trúc dữ liệu

Cấu trúc đề cập đến cách các biến trong một bộ dữ liệu liên quan với nhau. Đây là cấu trúc mà khoa học dữ liệu cần biết. Xu hướng, được đề cập ở trên, nhấn mạnh cấu trúc tự nhiên của một bộ dữ liệu để dữ liệu bị biến dạng ít đại diện cho thế giới thực hơn dữ liệu không thiên lệch. Nhưng bản thân cấu trúc có nhiều dạng khác nhau: cấu trúc vĩ mô, cấu trúc vi mô...

*Định nghĩa: Cấu trúc<sup>1</sup> là một sự sắp xếp và tổ chức các yếu tố bên trong một vật hay hệ thống nào đó, hoặc các đối tượng, hệ thống tổ chức.*

- **Cấu trúc thượng tầng** đề cập đến khung được dựng lên để thu thập dữ liệu và tạo thành một bộ dữ liệu. Cấu trúc thượng tầng được tạo ra một cách có ý thức, có chủ ý và dễ nhìn thấy. Khi bộ dữ liệu được tạo, các quyết định phải được đưa ra để xác định chính xác các phép đo nào được ghi lại, được đo theo cách nào và được lưu trữ theo định dạng nào;

Định nghĩa: Cấu trúc dữ liệu<sup>2</sup> là định dạng tổ chức, quản lý và lưu trữ dữ liệu cho phép truy cập và sửa đổi hiệu quả.

- **Cấu trúc vĩ mô** liên quan đến định dạng của các biến. Thí dụ, độ chi tiết là một tính năng cấu trúc vĩ mô. Độ chi tiết đề cập đến lượng chi tiết được ghi lại trong bất kỳ thời gian đo lường nào đến phút gần nhất, giờ gần nhất hoặc đơn giản là phân biệt sáng, chiều và đêm chẳng hạn. Các quyết định về cấu trúc vĩ mô có tác động quan trọng đến lượng thông tin mà bộ dữ liệu mang theo. Nó có ảnh hưởng rất lớn đến độ phân giải của bất kỳ mô hình nào được xây dựng bằng bộ dữ liệu đó. Tuy nhiên, cấu trúc vĩ mô không phải là một phần của khung được dựng lên một cách có ý thức để giữ dữ liệu, mà là bản chất của các phép đo;
- **Cấu trúc vi mô**, còn được gọi là cấu trúc tốt, mô tả các cách thức mà các biến đã được nắm bắt liên quan đến nhau. Chính dựa vào cấu trúc này mà mô hình hóa được dữ liệu. Một đánh giá cơ bản về trạng thái của cấu trúc vi mô có thể tạo thành một phần hữu ích của kiểm toán dữ liệu. Việc kiểm tra ngắn này là một đánh giá đơn giản về độ phức tạp của các biến tương quan giữa các biến. Người ta muốn đơn giản mà vẫn có mô hình dữ liệu tốt; tuy nhiên sự phức tạp sẽ tạo điều kiện để chính xác được mô hình dữ liệu.

---

<sup>1</sup> Structure: cấu trúc.

<sup>2</sup> Data structure: cấu trúc dữ liệu.



#### 2.4.3.6. Giai đoạn 6: Xây dựng PIE

Năm bước đầu tiên phần lớn yêu cầu đánh giá và hiểu dữ liệu có sẵn. Cần xem xét chi tiết về dữ liệu, xem đã thực hiện một số điều sau:

- Giúp xác định khả năng hoặc sự cần thiết của việc điều chỉnh hoặc chuyển đổi dữ liệu;
- Thiết lập những kỳ vọng hợp lý để đạt được một giải pháp;
- Xác định chất lượng chung hoặc tính hợp lệ của dữ liệu;
- Cho thấy sự liên quan của dữ liệu với nhiệm vụ trong tay.

Nhiều trong số các hoạt động này đòi hỏi phải áp dụng suy nghĩ và hiểu biết, hơn là các công cụ tự động. Tất nhiên, phần lớn đánh giá được trợ giúp bởi thông tin thu được bằng cách chuẩn bị dữ liệu và các công cụ khám phá khác, nhưng kết quả là thông tin ảnh hưởng đến quyết định về cách chuẩn bị và sử dụng dữ liệu.

Ở giai đoạn này, các giới hạn của dữ liệu đã được biết, ít nhất là trong chừng mực có thể. Các quyết định đã được đưa ra dựa trên thông tin được phát hiện. Các kĩ thuật hoàn toàn tự động để chuẩn bị dữ liệu hiện có.

Các quyết định được thực hiện cho đến nay xác định trình tự hoạt động. Trong môi trường tạo nên, bộ dữ liệu có thể thể hiện dưới dạng mà bất kỳ máy nào cũng có thể truy cập được. Trong thực tế, các kĩ thuật không có khả năng áp dụng chính xác như mô tả. Tổng hợp thông tin sẽ dễ dàng hơn, nếu người ta chuyển hóa các tệp dữ liệu sao cho phù hợp với công cụ.

#### *Vấn đề dữ liệu: Mẫu đại diện*

Đã từ lâu người ta đã thấy được vai trò của xác định lượng dữ liệu đối với việc xây dựng mô hình. Một nguyên lí của việc khai thác dữ liệu là đề cập tất cả các dữ liệu, mọi lúc mọi nơi. Đó là một nguyên tắc tốt và nếu có thể đạt được, thì đây là một mục tiêu đáng giá. Tuy nhiên, vì nhiều lí do, nó không phải là một giải pháp thực tế. Ngay cả khi cần kiểm tra càng nhiều dữ liệu càng tốt, khảo sát và mô hình hóa vẫn cần ít nhất ba bộ dữ liệu: (i) bộ dữ liệu huấn luyện; (ii) bộ dữ liệu kiểm tra; và (iii) bộ dữ liệu thực thi. Việc cải tiến tính năng có thể yêu cầu tập trung các trường hợp thể hiện một số tính năng cụ thể. Mức chi tiết của dữ liệu chỉ có thể được thỏa mãn nếu một tập hợp con dữ liệu được *trích xuất* từ bộ dữ liệu chính. Vì vậy, luôn luôn cần quyết định mức độ lớn của một bộ dữ liệu để phản ánh chính xác *cấu trúc tình hình* dữ liệu.

Trong trường hợp này, khi xây dựng PIE, điều quan trọng là bộ dữ liệu có cấu trúc tốt. Mọi nỗ lực phải được thực hiện để đảm bảo rằng chính PIE không đưa ra sự thiên vị. Nếu không, kiểm tra toàn bộ quần thể, có thể là không thể, thì không có cách nào chắc chắn 100% rằng bất kỳ mẫu cụ thể nào, trên thực tế, là dữ liệu đại

diện. Tuy nhiên, có thể là một số lượng được chỉ định ít hơn 100%, giả sử 99% hoặc 95%. Đây là những biện pháp chắc chắn cho phép lấy mẫu. Chọn một mức độ chắc chắn phù hợp là một quyết định theo kinh nghiệm.

#### *Vấn đề dữ liệu: Giá trị phạm trù*

Các danh mục được đánh số là số, hay được gán số thích hợp. Ngay cả khi kết thúc việc chuẩn bị dữ liệu, các dữ liệu phạm trù sẽ được mô hình hóa thành các giá trị phân loại; chúng vẫn còn đánh số để ước tính các giá trị còn thiếu. Đó là một *trật tự* thực sự tồn tại và được phản ánh trong các phép đo phân loại. Khi xây dựng các mô hình dự đoán hoặc suy luận, điều quan trọng là thứ tự tự nhiên của các giá trị phân loại phải được bảo tồn trong chừng mực có thể.

Thay đổi *thứ tự* tự nhiên này là áp đặt một cấu trúc. Ngay cả việc áp đặt một cấu trúc ngẫu nhiên cũng làm mất thông tin do tiến hành phép đo phân loại. Nếu nó không phải là ngẫu nhiên, tình hình tồi tệ hơn vì nó giới thiệu một mô hình không có trong thực tế.

*Phương pháp số* phụ thuộc hoàn toàn vào cấu trúc của bộ dữ liệu. Trong một tập hợp dữ liệu số/phân loại hỗn hợp, các giá trị số được sử dụng để phản ánh thứ tự của chúng vào các lớp. Đây là phương pháp thành công nhất, vì các giá trị số có khoảng cách về thứ tự và độ lớn. Trong các bộ dữ liệu toàn diện, điều này cho phép phục hồi hợp lý thứ tự phù hợp. Trong thực tế, cũng có thể chuyển đổi một biến thực sự là số thành giá trị phân loại và xem thứ tự chính xác theo các thứ tự phân loại.

Các bộ dữ liệu với các phép đo phân loại gặp vấn đề nhỏ: phục hồi thứ tự thích hợp của các phân loại. Vấn đề là không có các biến số trong bộ dữ liệu, các giá trị được phục hồi không được neo vào các hiện tượng trong thế giới thực. Số này là tốt cho mô hình và trong thực tế đã tạo ra các mô hình hữu ích. Tuy nhiên, đó là một thực tế nguy hiểm khi sử dụng các thứ tự số để suy ra bất cứ điều gì tuyệt đối về mức độ có ý nghĩa của một đối tượng được phân loại. Các mối quan hệ của các biến, cái này với cái khác, giữ đúng, nhưng không được neo trở lại thế giới thực như các giá trị số.

Điều quan trọng cần lưu ý là không có *phương pháp tự động* nào có thể tự động xếp thứ tự như thế giới thực yêu cầu. Bất kỳ bộ dữ liệu nào cũng là một sự phản ánh của thế giới thực. Vì vậy, bất cứ khi nào có thể, các giá trị phân loại được sắp xếp phải được đặt theo thứ tự thích hợp của chúng là giá trị thứ tự. Tuy nhiên, vì thường xảy ra trường hợp khi mô hình hóa dữ liệu không có chuyên gia về lĩnh vực hoặc không có xếp hạng thứ tự rõ ràng, nên người ta đành sử dụng các kỹ thuật tự động.

### *Vấn đề dữ liệu: Chuẩn hóa*

Một số loại chuẩn hóa rất hữu ích khi mô hình hóa. Chuẩn hóa đã được sử dụng trong cơ sở dữ liệu quan hệ. Đưa dữ liệu vào các dạng thông thường khác nhau của nó trong cơ sở dữ liệu đòi hỏi phải sử dụng nhiều *bảng* quan hệ. Các bảng cần tuân theo chuẩn 1NF đến 5NF.

*Định nghĩa: Chuẩn hóa cơ sở dữ liệu<sup>1</sup> là một phương pháp khoa học để phân tách một bảng có cấu trúc phức tạp thành những bảng có cấu trúc đơn giản theo những qui luật không làm mất thông tin.*

Việc chuẩn hóa cơ sở dữ liệu sẽ làm giảm bớt sự dư thừa và loại bỏ những sự cố mâu thuẫn về dữ liệu, tiết kiệm được không gian lưu trữ. Một số dạng chuẩn hóa dữ liệu thông dụng là (i) dạng chuẩn 1NF; (ii) dạng chuẩn 2NF; (iii) dạng chuẩn 3NF<sup>2</sup>; (iv) dạng chuẩn Boyce-Codd BCNF. Chuẩn hóa dữ liệu còn có ý nghĩa khác ngoài việc lưu trữ dữ liệu trên các máy tính. Đối với các văn bản, việc chuẩn hóa dữ liệu có thể làm cho văn bản trở nên dễ đọc hơn, không vướng vào những trường hợp về hiển thị.

Một số công cụ, chẳng hạn như mạng nơ ron, yêu cầu chuẩn hóa phạm vi. Các công cụ khác không yêu cầu chuẩn hóa, nhưng được lợi từ việc có dữ liệu chuẩn hóa. Người làm khoa học dữ liệu nên kiểm soát quá trình chuẩn hóa.

### *Vấn đề dữ liệu: Giá trị thiếu và giá trị trống*

Xử lý các giá trị *thiếu* và *trống* là rất quan trọng. Không may là chưa có kỹ thuật tự động để phân biệt giữa các giá trị bị thiếu và trống. Nếu được thực hiện, công cụ khai thác phải phân biệt thủ công, nhập mã phân loại cho dù giá trị bị thiếu hay trống. Nếu việc phát hiện giá trị thiếu hay trống có thể được thực hiện, có thể tạo ra kết quả hữu ích. Thông thường có thể thực hiện được điều này hoặc thực hiện với tỉ lệ nào đó. Các giá trị trống và thiếu chỉ đơn giản là phải được đối xử như nhau.

Tất cả các công cụ mô hình hóa đều có một số phương tiện xử lý các giá trị bị thiếu, ngay cả khi nó bỏ qua mọi trường hợp có chứa giá trị bị thiếu. Các chiến lược khác bao gồm gán một số giá trị cố định cho tất cả các giá trị bị thiếu của một biến cụ thể hoặc xây dựng một số ước tính về giá trị còn thiếu, dựa trên các giá trị của các biến khác. Có vấn đề với tất cả các cách tiếp cận này vì mỗi cách thể hiện một số hình thức thỏa hiệp.

Khi các giá trị bị thiếu được thay thế, thông tin về mẫu của các giá trị bị thiếu sẽ bị mất. Một phân loại sơ bộ sẽ được tạo ra để nắm bắt thông tin về giá trị cho mỗi

---

<sup>1</sup> Database norm: chuẩn cơ sở dữ liệu.

<sup>2</sup> 3NF: the third Norm Form: dạng chuẩn 3.

mẫu giá trị bị thiếu. Chỉ sau khi thông tin này được nắm bắt, các giá trị thiếu mới được thay thế.

#### *Vấn đề dữ liệu: Dòng dịch chuyển*

Tại thời điểm này trong quá trình chuẩn bị, dữ liệu được hiểu, tăng cường, làm giàu, lấy mẫu đầy đủ, đánh số đầy đủ, chuẩn hóa theo hai chiều, có phạm vi và phân phối, và cân bằng. Nếu bộ dữ liệu là một *chuỗi dịch chuyển*, mà chuỗi thời gian là phổ biến nhất, thì bộ dữ liệu được xử lý bằng các kỹ thuật chuẩn bị chuyên biệt khác nhau.

Hành động quan trọng nhất ở đây mà không thể tự động hóa một cách an toàn là yêu cầu kiểm tra dữ liệu của người dùng. Giảm dần chuỗi dịch chuyển có thể là một hoạt động hủy hoại đối với nội dung thông tin nếu trên thực tế dữ liệu không có xu hướng thực sự. Vậy nên thận trọng; người dùng phải đưa ra một số quyết định cần tuân theo và/hoặc sử dụng bộ lọc để chuẩn bị dữ liệu.

Lúc này, PIE được xây dựng. Đã có (i) một số dạng chương trình máy tính, phương trình toán học; (ii) dữ liệu thô; (iii) các biến. Mỗi biến được xem xét tách biệt với mối quan hệ của nó với các biến khác.

#### *Vấn đề bộ dữ liệu: Giảm chiều rộng*

Các bộ dữ liệu để khai thác có thể được coi là được tạo từ một bảng hai chiều với các cột biểu thị các phép đo biến và các hàng biểu thị các thể hiện hoặc các bản ghi. Chiều rộng mô tả số lượng cột, trong khi độ sâu mô tả số lượng hàng.

Một trong những vấn đề khó khăn nhất mà một người làm khoa học dữ liệu là *độ rộng* của dữ liệu. Nhiều biến có lẽ mang nhiều thông tin hơn. Nhưng quá nhiều biến số có thể sẽ quá tải, về (i) năng lực tính toán; (ii) dung lượng bộ nhớ. Hầu hết các thuật toán có nhiều cách khác nhau để giảm độ phức tạp tổ hợp của giao tác mô hình hóa, nhưng quá nhiều biến số có thể ngăn cản bất kỳ phương thức nào.

Tuy nhiên khoa học dữ liệu có thể muốn giảm số lượng cột dữ liệu trong một bộ dữ liệu, nhưng không làm giảm nội dung thông tin của nó.

#### *Vấn đề về bộ dữ liệu: giảm độ sâu*

*Độ sâu* không có tác động quá tải như *độ rộng* có thể gây nên. Tuy nhiên, độ sâu liên quan đến thời gian khai thác dữ liệu. Giải pháp là sử dụng tập dữ liệu đại diện của toàn thể dữ liệu. Tuy nhiên cần phải đảm bảo rằng tập hợp con của dữ liệu được mô hình hóa trên thực tế phản ánh tất cả các mối quan hệ tồn tại trong bộ dữ liệu đầy đủ. Điều này đòi hỏi một cái nhìn khác về lấy mẫu. Lần này, việc lấy mẫu phải xem xét các tương tác giữa các biến, không chỉ xem xét sự biến đổi của các biến riêng lẻ.

### *Vấn đề tập hợp dữ liệu/khảo sát dữ liệu: Các đa tạp có dạng tốt và đa tạp có dạng không tốt*

Đây thực sự là bước khảo sát dữ liệu đầu tiên cũng như bước chuẩn bị dữ liệu cuối cùng. Khảo sát dữ liệu liên quan đến việc quyết định những gì trong bộ dữ liệu trước khi lập mô hình. Tuy nhiên, nó cũng là phần cuối cùng của việc chuẩn bị dữ liệu bởi vì nếu có vấn đề với hình dạng của đa tạp, có thể xử lý dữ liệu để cải thiện một số trong số chúng.

*Định nghĩa: Một đa tạp<sup>1</sup> tôpô  $n$  chiều là một không gian tôpô mà mỗi điểm có lân cận đồng phôi với tập con mở của  $R^n$ ; nói một cách khác, là không gian tôpô tách được với mỗi điểm của nó có một lân cận đồng phôi với một tập mở trong không gian Oclit. Đa tạp chính là khái niệm toán học mở rộng của đường và mặt.*

Cuộc khảo sát không liên quan đến nội dung dữ liệu, nhưng thông tin về xử lý sẽ giúp ích cho việc mô hình hóa. Là bước cuối cùng trong việc chuẩn bị dữ liệu, cái nhìn đa dạng này sẽ tìm cách xác định xem có vấn đề nào có thể được loại bỏ.

#### *2.4.3.7. Giai đoạn 7: Khảo sát dữ liệu*

Khảo sát dữ liệu sẽ (i) kiểm tra; và (ii) báo cáo về các tính chất chung của đa tạp trong không gian trạng thái. Theo nghĩa đen, nó tạo ra một bản đồ các thuộc tính của đa tạp, tập trung vào các thuộc tính mà người khai thác thấy hữu ích và quan trọng nhất.

Người lập mô hình quan tâm đến việc *biết nhiều tính năng*, chẳng hạn như (i) mật độ tương đối của các điểm trong không gian trạng thái; (ii) các cụm xuất hiện tự nhiên; (iii) các biến dạng không chính xác và nơi chúng xảy ra; (iv) các khu vực có độ thưa tương đối cụ thể; (v) cách xác định mức độ đa dạng...

Vì thông tin được phát hiện khi khảo sát dữ liệu ảnh hưởng đến cách chuẩn bị dữ liệu, nó trở nên một phần của quá trình chuẩn bị dữ liệu.

#### *2.4.3.8. Giai đoạn 8: Mô hình hóa dữ liệu*

Mục đích của việc chuẩn bị và khảo sát là để hiểu dữ liệu. Thông thường, sự hiểu biết cần phải được biến thành một mô hình *chủ động* hoặc *thụ động, bị động*. Như với khảo sát dữ liệu, mô hình hóa là một chủ đề quá rộng. Một số thiếu sót và vấn đề chỉ xuất hiện khi cố gắng mô hình hóa. Yêu cầu về mô hình hóa dữ liệu thúc đẩy các nỗ lực chuẩn bị dữ liệu một cách hiệu quả trong nỗ lực cải thiện các vấn đề. Mô hình hóa cũng có một số vai trò trong việc chuẩn bị dữ liệu.

---

<sup>1</sup> Manifolds: đa tạp.

Liên quan đến việc chuẩn bị dữ liệu cho quá trình khoa học dữ liệu, có một số nhận xét sau:

- Rõ ràng, một số chuẩn bị dữ liệu *tối thiểu* phải được thực hiện cho bất kỳ công cụ mô hình hóa. Thí dụ, mạng nơ ron yêu cầu tất cả các đầu vào phải được đánh số và phạm vi được chuẩn hóa;
- Các *kỹ thuật khác* đòi hỏi sự chuẩn bị tối thiểu khác. Xác định lợi ích đạt được, nếu thực hiện các bước bổ sung trên mức tối thiểu. Hầu hết các công cụ được mô tả là có thể tìm hiểu các mối quan hệ phức tạp giữa các biến. Đây là mục đích của việc chuẩn bị dữ liệu: chuyển đổi các bộ dữ liệu sao cho nội dung thông tin của chúng được tiếp xúc tốt nhất với công cụ khai thác. Một điều rất quan trọng là nếu không có điều tốt nào có thể được thực hiện trong một bộ dữ liệu cụ thể, thì ít nhất không xảy ra bất kì sai sót nào;
- *So sánh hiệu suất* của cùng một công cụ trên cùng một bộ dữ liệu ở cả hai trạng thái (i) được chuẩn bị tối thiểu; (ii) được chuẩn bị đầy đủ của chúng, nhằm đưa ra một dấu hiệu công bằng về những gì có thể được mong đợi;
- Có một số bộ dữ liệu chưa được cải thiện về *tỉ lệ lỗi* dự đoán. Trong những trường hợp này, điều quan trọng cần lưu ý là không có sự xuống cấp, nghĩa là ít nhất không có tác hại. Thông thường có sự cải thiện trong một số trường hợp, một số lượng nhỏ, trong những trường hợp khác thì nhiều hơn. Vì *hiệu suất thực tế* phụ thuộc vào dữ liệu, nên khó để nói hiệu ứng nào sẽ được tìm thấy trong bất kỳ trường hợp cụ thể nào. Tỉ lệ lỗi cũng chịu ảnh hưởng bởi loại phân loại và độ chính xác có thể bị ảnh hưởng rất khác nhau khi sử dụng cùng một mô hình và cùng một bộ dữ liệu. Khi tỉ lệ lỗi được xác định, có sự cải thiện đáng kể về hiệu suất mô hình khi các mô hình được xây dựng và thực hiện trên dữ liệu đã chuẩn bị;
- Còn nhiều thứ đối với chuẩn bị dữ liệu, như *giảm chiều, giảm độ rộng, độ sâu...*, hơn là chỉ cải thiện tỉ lệ lỗi. Giảm biến thường tăng thời gian khai thác từ 10 đến 100 lần so với dữ liệu chưa chuẩn bị. Hơn nữa, một số bộ dữ liệu còn thô và bị bóp méo trước khi chuẩn bị, đến mức chúng không thể sử dụng được. Các kỹ thuật chuẩn bị dữ liệu làm cho dữ liệu ít nhất có thể sử dụng được, đó là một mức đáng kể trong chính dữ liệu. Ít nhất là người ta tin tưởng vào dữ liệu trước khi mô hình hóa bắt đầu. *Quan điểm* này có thể có giá trị hơn bất kỳ cải thiện về hiệu suất mô hình. Đây là nơi chuẩn bị của người làm khoa học dữ liệu, thông qua cái nhìn sâu sắc đối với dữ liệu, hơn là chú trọng quá vào xây dựng các mô hình tốt hơn. Và hiệu quả của điều đó là không thể định lượng;

- Việc áp dụng các kĩ thuật (i) giảm tỉ lệ lỗi trong mô hình; (ii) giảm thời gian xây dựng mô hình; (iii) có cái nhìn sâu sắc về dữ liệu là quan trọng.

## 2.5. Kết luận

Chương trên đã đề cập qui trình làm việc với dữ liệu, để xây dựng được mô hình trên dữ liệu. Qui trình này được gọi là *quá trình khoa học dữ liệu*, gồm sáu bước.

Liên quan đến quá trình khoa học dữ liệu, có nhiệm vụ (i) thăm dò dữ liệu; (ii) chuẩn bị dữ liệu. Nội dung này được trình bày trong các phần trên, như mục quan trọng đối với quá trình khoa học dữ liệu.

Ngôn ngữ sử dụng để thể hiện quá trình khoa học dữ liệu là Python. Vậy nên việc chuẩn bị hạ tầng cho ngôn ngữ này cũng là khía cạnh cần quan tâm đối với người làm khoa học dữ liệu.

## Các câu hỏi

1. Hãy phân tích từng pha của quá trình khoa học dữ liệu ?
2. So sánh các thuật ngữ cleaning và cleansing khi làm sạch dữ liệu ?
3. Sử dụng Jupyter hay Colab để thực hiện bài toán học máy phân loại bằng cây quyết định, với dữ liệu về kết quả học tập của sinh viên trong trường ?
4. Sử dụng Jupyter hay Colab để thực hiện bài toán học máy để thực hiện mô hình hồi qui tuyến tính đối với dữ liệu về dân số Việt Nam ?
5. Sử dụng Jupyter hay Colab để thực hiện bài toán học sâu với mạng tích chập, với dữ liệu cần phân loại ?
6. Sử dụng Jupyter hay Colab để thực hiện bài toán học sâu về dịch đoạn văn bản tiếng Việt sang tiếng Anh ?



