ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

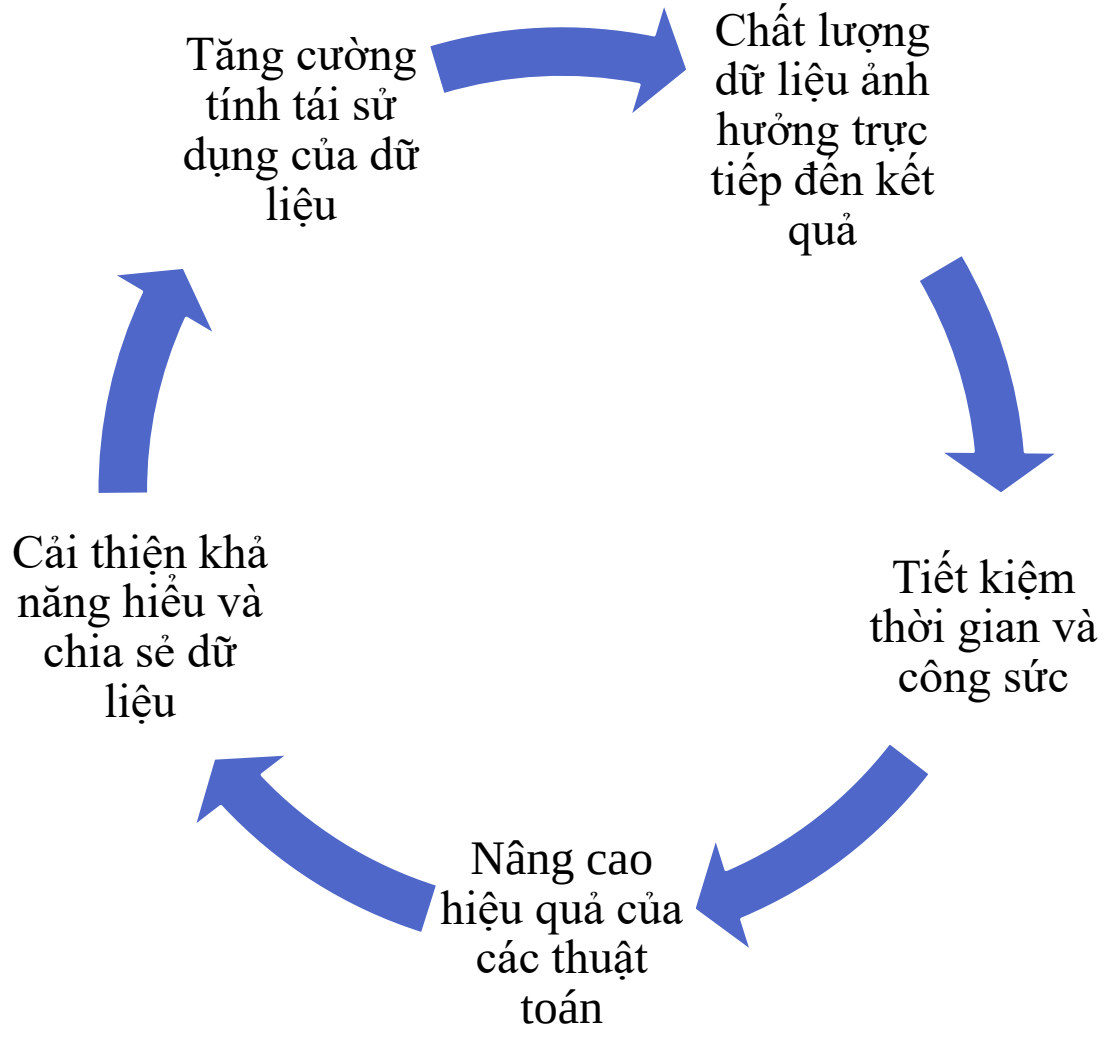
CHƯƠNG 3
Chuẩn bị dữ liệu

Biên soạn: ThS. Nguyễn Thị Anh Thư

Nội dung

1. Tổng quan
2. Các loại dữ liệu
3. Khám phá dữ liệu
4. Làm sạch dữ liệu
5. Chuyển đổi dữ liệu
6. Ghi nhãn dữ liệu
7. Chia tập dữ liệu
8. Bài tập
9. Tổng kết

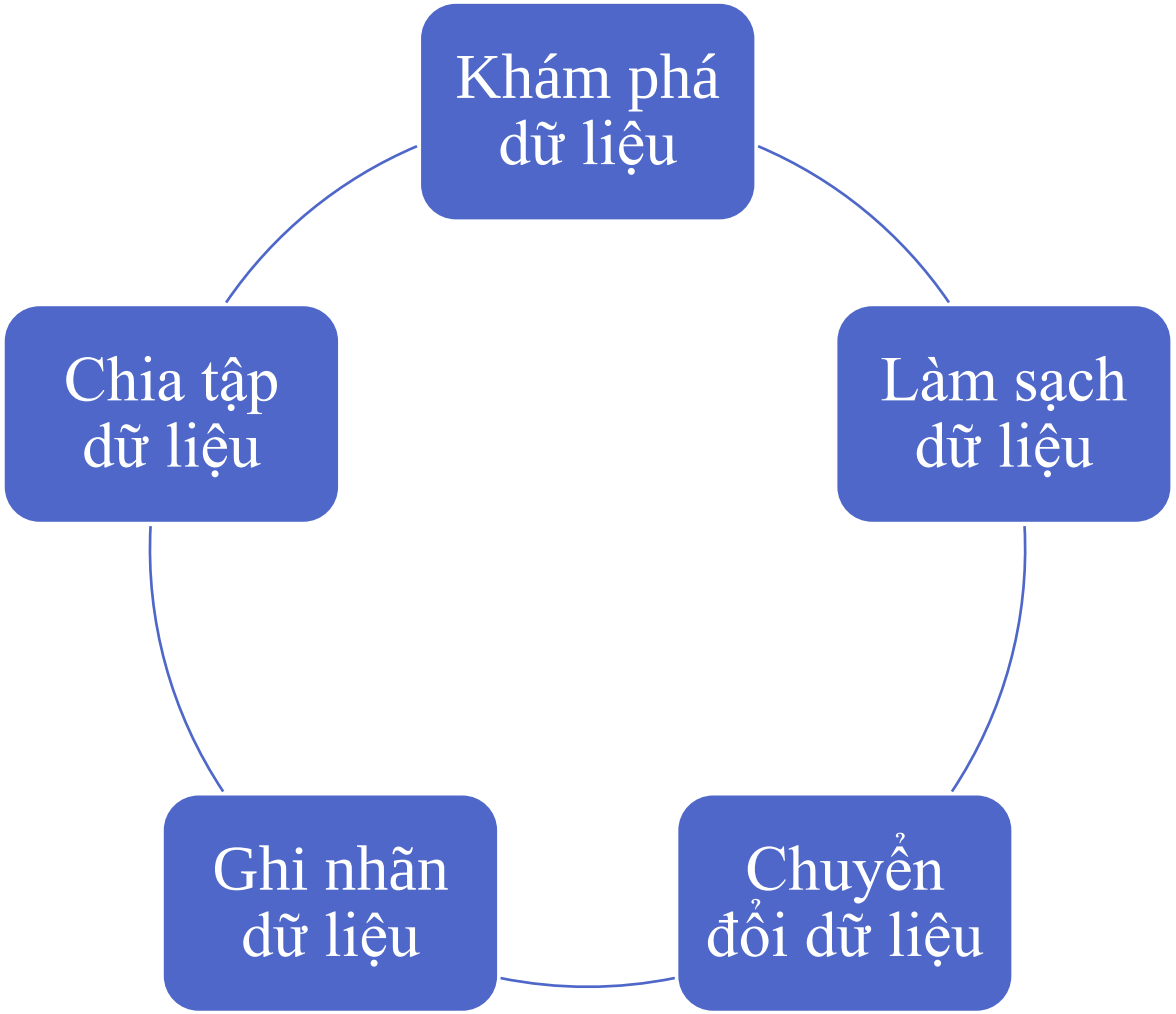
1. Tổng quan



Chuẩn bị dữ liệu là một bước quan trọng giúp nâng cao chất lượng dữ liệu, tiết kiệm thời gian và công sức, và cải thiện hiệu quả của các dự án dữ liệu lớn.

Quá trình chuẩn bị dữ liệu có thể tốn nhiều thời gian và công sức, nhưng đây là bước quan trọng để đảm bảo tính chính xác và hiệu quả của mô hình học máy.

1. Tổng quan


Khám phá dữ liệu
Làm sạch dữ liệu
Chuyển đổi dữ liệu
Ghi nhãn dữ liệu
Chia tập dữ liệu

2. Các loại dữ liệu

Dữ liệu có cấu trúc (Structured data)	Dữ liệu bán cấu trúc (Semi-structured data)	Dữ liệu phi cấu trúc (Unstructured data)
Cơ sở dữ liệu quan hệ: MySQL, PostgreSQL, Oracle, v.v. **Cơ sở dữ liệu NoSQL:** MongoDB, Cassandra, HBase, v.v. **Dữ liệu bảng:** Bảng dữ liệu trong Excel, Google Sheets, v.v. **Dữ liệu GIS:** Dữ liệu bản đồ và vị trí **Dữ liệu giao dịch:** Lịch sử mua hàng, dữ liệu clickstream, v.v.	**Dữ liệu bảng tính:** Excel, CSV, TSV, v.v. **Dữ liệu XML:** Dữ liệu được đánh dấu theo định dạng XML **Dữ liệu JSON:** Dữ liệu được định dạng JSON **Email:** Email có cấu trúc (ví dụ: với các tiêu đề và phần nội dung) **Dữ liệu thư mục:** Cấu trúc thư mục và tệp trên máy tính	**Văn bản:** Email, bài đăng trên mạng xã hội, bài báo, sách, v.v. **Hình ảnh:** JPEG, PNG, GIF, v.v. **Âm thanh:** MP3, WAV, AAC, v.v. **Video:** MP4, MOV, AVI, v.v. **Dữ liệu cảm biến:** Dữ liệu từ các thiết bị IoT, ghi chép y tế, v.v.
Ví dụ: Hồ sơ khách hàng trong cơ sở dữ liệu CRM	Ví dụ: Bảng tính Excel với danh sách khách hàng	Ví dụ: Bài đăng trên blog (văn bản)

Tên	Email	Số điện thoại	Địa chỉ
Nguyễn Văn A	[đã xoá địa chỉ email]	0123456789	123 Nguyễn Trãi, TP.HCM
Lê Thị B	[đã xoá địa chỉ email]	0987654321	456 Lê Văn Quới, TP.HCM
Trần Văn C	[đã xoá địa chỉ email]	0123456789	789 Phạm Văn Chí, TP.HCM

2. Các loại dữ liệu

Bảng tính Excel này là một ví dụ về dữ liệu bán cấu trúc.

- Bảng tính này có cấu trúc rõ ràng với các hàng và cột được xác định. Dữ liệu được định dạng nhất quán và có thể được sử dụng để phân tích, lọc và sắp xếp.
- Tuy nhiên, bảng tính này cũng có một trường không có cấu trúc là "Ghi chú" (không được hiển thị trong ví dụ).

2. Các loại dữ liệu

- Ranh giới giữa dữ liệu bán cấu trúc và dữ liệu có cấu trúc có thể không rõ ràng.
- Một số loại dữ liệu có thể thuộc nhiều loại, tùy thuộc vào cách thức lưu trữ và sử dụng.
- *Ví dụ*: Dữ liệu XML có thể được coi là dữ liệu bán cấu trúc hoặc dữ liệu có cấu trúc.

XML

```xml
<danh_ba>
  <nguoi_lien_he>
    <ho>Nguyen</ho>
    <ten>Van A</ten>
    <so_dien_thoai>0123456789</so_dien_thoai>
    <email>van.nguyen@example.com</email>
  </nguoi_lien_he>
  <nguoi_lien_he>
    <ho>Tran</ho>
    <ten>Thi B</ten>
    <so_dien_thoai>0987654321</so_dien_thoai>
    <email>thi.tran@example.com</email>
  </nguoi_lien_he>
</danh_ba>
```

Dữ liệu XML bán cấu trúc có thể được chuyển đổi sang dạng bảng trong cơ sở dữ liệu quan hệ.

2. Các loại dữ liệu

Công cụ và thư viện hỗ trợ - Python

- ***Pandas***: Thư viện mạnh mẽ để thao tác dữ liệu dạng bảng, xử lý dữ liệu thiếu, lọc, sắp xếp, tổng hợp và trực quan hóa.
- ***NumPy***: Cung cấp các công cụ cho tính toán khoa học, thao tác với mảng dữ liệu đa chiều hiệu quả.
- ***SciPy***: Bao gồm các module cho thống kê, tối ưu hóa, tích hợp số và giải phương trình vi phân.
- ***Matplotlib***: Thư viện trực quan hóa dữ liệu phổ biến, tạo ra các biểu đồ, đồ thị và hình ảnh chất lượng cao.
- ***Seaborn***: Cung cấp các giao diện lập trình đơn giản và đẹp mắt để tạo ra các đồ thị thống kê với Matplotlib.
- ***Dask***: Thư viện cho phép xử lý dữ liệu lớn hiệu quả trên các cụm máy tính.

2. Các loại dữ liệu

Công cụ và thư viện hỗ trợ - R

- ***data.frame***: Cấu trúc dữ liệu cơ bản để lưu trữ và thao tác dữ liệu dạng bảng.
- ***tidyverse***: Bộ sưu tập các gói thống nhất cho thao tác dữ liệu, phân tích thống kê và trực quan hóa.
- ***ggplot2***: Thư viện trực quan hóa dữ liệu mạnh mẽ, tạo ra các đồ thị theo nguyên tắc "grammar of graphics".
- ***plyr***: Cung cấp các hàm để thao tác dữ liệu theo nhóm, tóm tắt và chuyển đổi dữ liệu.
- ***caret***: Hỗ trợ các chức năng cho học máy và mô hình hóa thống kê.

2. Các loại dữ liệu

Công cụ và thư viện hỗ trợ - Java

- ***Apache Commons Collections***: Cung cấp các bộ sưu tập dữ liệu và công cụ thao tác dữ liệu.
- ***Apache Commons Lang***: Cung cấp các hàm tiện ích cho thao tác chuỗi, ngày tháng và các kiểu dữ liệu khác.
- ***Apache POI***: Hỗ trợ đọc và ghi dữ liệu từ các tệp Microsoft Office.
- ***Weka***: Cung cấp các công cụ cho học máy và phân tích dữ liệu.

2. Các loại dữ liệu

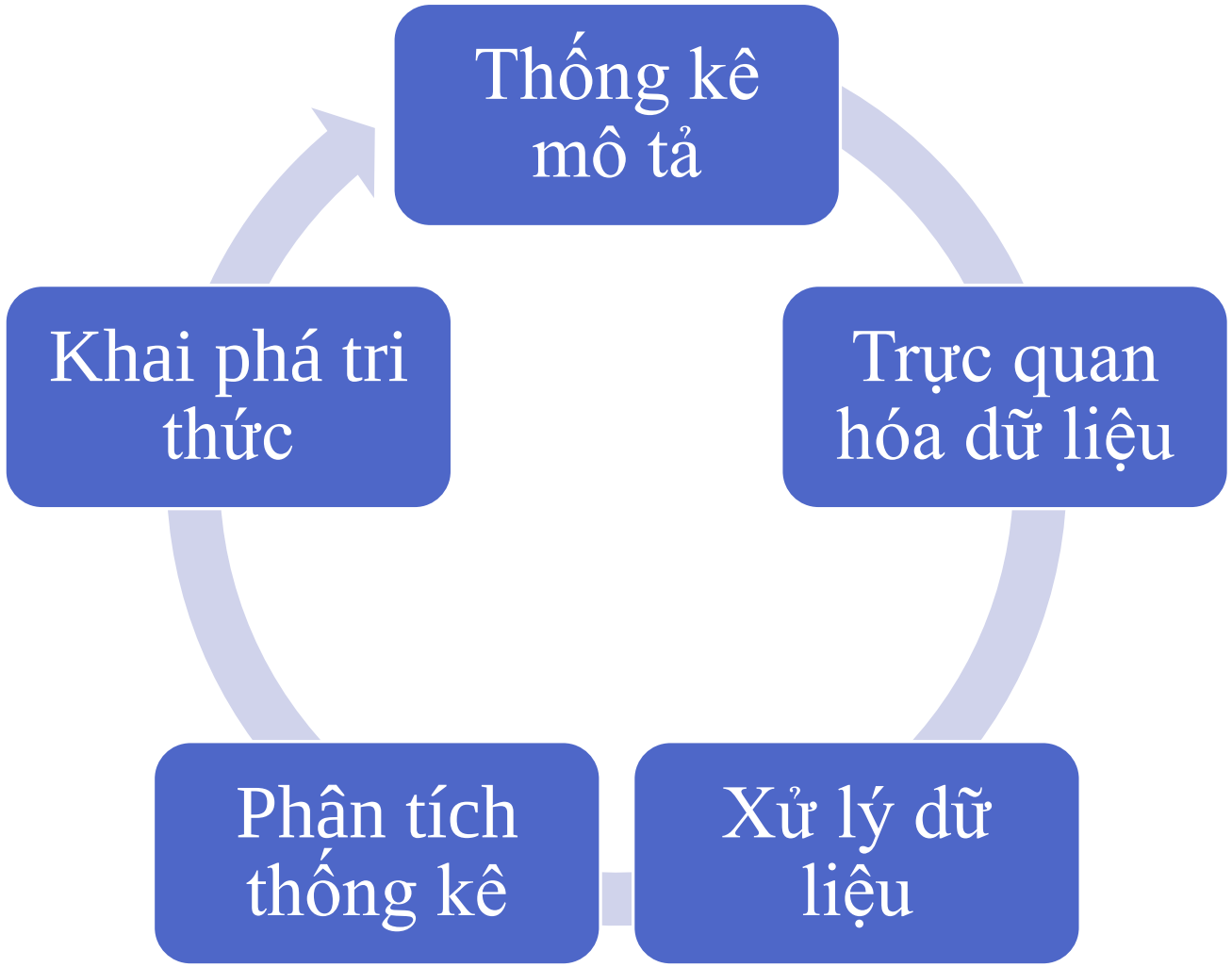
Công cụ và thư viện hỗ trợ - Scala

- ***Apache Spark***: Khung xử lý dữ liệu song song cho các tập dữ liệu lớn.
- ***MLlib***: Thư viện học máy cho Spark, bao gồm các thuật toán phân loại, hồi quy và cụm.
- ***DataFrame***: Cấu trúc dữ liệu dạng bảng tương tự như Pandas trong Python.
- ***Breeze***: Cung cấp các hàm cho tính toán khoa học và thao tác với mảng dữ liệu.

Ngoài ra

- ***SQL***: Ngôn ngữ truy vấn cơ sở dữ liệu để truy xuất, lọc và thao tác dữ liệu trong cơ sở dữ liệu.
- ***NoSQL***: Các cơ sở dữ liệu NoSQL như MongoDB và Cassandra cung cấp các API để truy xuất và thao tác dữ liệu phi cấu trúc.

3. Khám phá dữ liệu



Exploratory Data Analysis (EDA)

- Là một bước quan trọng.
- Giúp hiểu rõ hơn về tập dữ liệu và xác định các mẫu, xu hướng tiềm ẩn.

3. Khám phá dữ liệu

Thống kê mô tả

- Tính toán các thống kê cơ bản như trung bình, trung vị, độ lệch chuẩn, tứ phân vị, v.v.
  - **NumPy** là lựa chọn tốt cho các thao tác mảng dữ liệu hiệu quả.
  - **Pandas** là lựa chọn tốt cho việc phân tích dữ liệu với DataFrame.
  - **SciPy** cung cấp các hàm thống kê nâng cao hơn.
  - **Statistics** cung cấp các hàm thống kê cơ bản với cú pháp đơn giản.
- Ví dụ: NumPy
  - np.mean() - Tính trung bình
  - np.median() - Tính trung vị
  - np.std() - Tính độ lệch chuẩn
  - np.percentile() - Tính tứ phân vị

3. Khám phá dữ liệu

Thống kê mô tả

- Phân tích phân phối dữ liệu bằng cách sử dụng biểu đồ histogram, boxplot, v.v.
  - **Seaborn**: Thư viện trực quan hóa dữ liệu phổ biến nhất trong Python, cung cấp nhiều hàm vẽ biểu đồ nâng cao.
  - **Matplotlib**: Thư viện trực quan hóa dữ liệu cơ bản trong Python, cung cấp nhiều hàm vẽ biểu đồ cơ bản.
  - **Plotly**: Thư viện trực quan hóa dữ liệu tương tác, cho phép tạo biểu đồ có thể xoay, thu phóng và di chuyển.
- Ví dụ: Seaborn
  - displot(data, kind="hist"): Vẽ histogram đơn giản.
  - displot(data, kind="kde"): Vẽ mật độ xác suất (KDE) thay vì histogram.
  - displot(data, kind="rug"): Vẽ "thảm" dữ liệu bên dưới histogram.

3. Khám phá dữ liệu

Thống kê mô tả

- Xác định các giá trị ngoại lai (outlier) có thể ảnh hưởng đến kết quả phân tích.
  - **Phương pháp thống kê**: Phương pháp này sử dụng các thống kê như độ lệch chuẩn và khoảng tứ phân vị để xác định các giá trị nằm ngoài phạm vi bình thường.
  - **Phương pháp dựa trên khoảng cách**: Phương pháp này xác định các giá trị có khoảng cách lớn nhất so với các giá trị khác trong tập dữ liệu.
  - **Phương pháp dựa trên mật độ**: Phương pháp này xác định các giá trị nằm trong các khu vực có mật độ dữ liệu thấp.
- Ví dụ:
  - NumPy: np.std(), np.percentile()
  - SciPy: scipy.stats.zscore(), scipy.stats.boxplot()
  - Pandas: df.describe()

3. Khám phá dữ liệu

Trực quan hóa dữ liệu

- Sử dụng các biểu đồ như scatter plot, heatmap, line plot, v.v. để thể hiện mối quan hệ giữa các biến.
  - **Matplotlib**: Phù hợp cho việc vẽ đồ thị đơn giản và tùy chỉnh cao.
  - **Seaborn**: Phù hợp cho việc vẽ đồ thị đẹp mắt và dễ sử dụng.
  - **Plotly**: Phù hợp cho việc vẽ đồ thị tương tác và chia sẻ trực tuyến.
  - **Bokeh**: Phù hợp cho việc vẽ đồ thị tương tác và tích hợp với các ứng dụng web.
- Ví dụ: Matplotlib
  - Scatter plot: plt.scatter(x, y)
  - Heatmap: plt.imshow(data)
  - Line plot: plt.plot(x, y)

3. Khám phá dữ liệu

Trực quan hóa dữ liệu

- Tạo các dashboard trực quan để theo dõi các chỉ số quan trọng và khám phá các mẫu dữ liệu theo thời gian.
  - **Mức độ dễ sử dụng**: Streamlit là lựa chọn tốt nhất cho người mới bắt đầu.
  - **Tính linh hoạt**: Plotly và Dash cung cấp nhiều tính năng và khả năng tùy chỉnh.
  - **Khả năng tương tác**: Bokeh và Streamlit cung cấp khả năng tương tác thời gian thực.
  - **Loại biểu đồ**: Matplotlib hỗ trợ nhiều loại biểu đồ khác nhau.
- Ví dụ: streamlit
  - st.line_chart(data): Tạo biểu đồ
  - st.table(data): Tạo bảng

3. Khám phá dữ liệu

Trực quan hóa dữ liệu

- Sử dụng các kỹ thuật như dimensionality reduction để giảm bớt số lượng biến và dễ dàng trực quan hóa dữ liệu.
  - Hiểu rõ các phương pháp giảm chiều trước khi sử dụng.
  - Lựa chọn phương pháp phù hợp với dữ liệu và mục tiêu.
  - Thử nghiệm và so sánh các phương pháp khác nhau để đạt hiệu quả tốt nhất.
- Ví dụ:
  - Hàm *np.linalg.svd* trong *NumPy* thực hiện Singular Value Decomposition (SVD).
  - Hàm *sklearn.decomposition.PCA* trong *Scikit-learn* thực hiện Phân tích thành phần chính (PCA).
  - Hàm *sklearn.manifold.TSNE* trong *Scikit-learn* thực hiện T-SNE.
  - Hàm *tensorflow.keras.layers.Embedding* trong *TensorFlow* thực hiện dimensionality reduction cho dữ liệu văn bản.

3. Khám phá dữ liệu

Xử lý dữ liệu

- Làm sạch dữ liệu cơ bản bằng cách xử lý các giá trị thiếu, nhiễu và không nhất quán.
- **Xử lý giá trị thiếu**
  - pandas.isnull(): Kiểm tra giá trị NaN trong DataFrame.
  - df.dropna(): Xóa các hàng có giá trị NaN.
  - df.fillna(): Thay thế giá trị NaN bằng giá trị khác (số trung bình, giá trị trung vị, v.v.).
  - impute.SimpleImputer(): Thư viện từ scikit-learn để thay thế giá trị thiếu bằng nhiều phương pháp khác nhau.

3. Khám phá dữ liệu

Xử lý dữ liệu

- Làm sạch dữ liệu cơ bản bằng cách xử lý các giá trị thiếu, nhiễu và không nhất quán.
- **Xử lý nhiễu**
  - df.describe(): Thống kê tóm tắt về dữ liệu, giúp xác định các giá trị ngoại lai.
  - df.boxplot(): Tạo biểu đồ boxplot để trực quan hóa các giá trị ngoại lai.
  - df.replace(): Thay thế các giá trị ngoại lai bằng giá trị khác.
  - StandardScaler(): Thư viện từ scikit-learn để chuẩn hóa dữ liệu, giúp giảm thiểu ảnh hưởng của nhiễu.

3. Khám phá dữ liệu

Xử lý dữ liệu

- Làm sạch dữ liệu cơ bản bằng cách xử lý các giá trị thiếu, nhiễu và không nhất quán.
- **Xử lý dữ liệu không nhất quán**
  - df.dtypes: Kiểm tra kiểu dữ liệu của các cột.
  - df.astype(): Chuyển đổi kiểu dữ liệu của các cột.
  - df.replace(): Thay thế các giá trị không nhất quán bằng giá trị hợp lệ.
  - dict(): Tạo dictionary để ánh xạ các giá trị không nhất quán thành giá trị hợp lệ.

3. Khám phá dữ liệu

Xử lý dữ liệu

➢Chuyển đổi dữ liệu sang dạng phù hợp cho việc phân tích, ví dụ như mã hóa dữ liệu categorical.

Pandas	NumPy
• df.fillna(value): Thay thế giá trị thiếu bằng giá trị được chỉ định. • df.replace(to_replace, value): Thay thế giá trị cụ thể bằng giá trị khác. • df.astype(dtype): Chuyển đổi kiểu dữ liệu của cột sang kiểu dữ liệu mong muốn. • df.convert_dtypes(): Tự động chuyển đổi kiểu dữ liệu của cột sang kiểu phù hợp nhất. • df.dropna(axis=0, thresh=n): Loại bỏ các hàng có giá trị thiếu (axis=0) hoặc giữ lại các hàng có ít nhất n giá trị không thiếu (thresh=n).	• np.nan_to_num(x): Chuyển đổi giá trị NaN thành số. • np.where(condition, x, y): Thay thế các phần tử trong mảng dựa trên điều kiện. • np.array(data, dtype): Chuyển đổi dữ liệu sang dạng mảng NumPy với kiểu dữ liệu mong muốn. • np.reshape(array, shape): Thay đổi hình dạng của mảng NumPy.

3. Khám phá dữ liệu

Xử lý dữ liệu

- Kết hợp các tập dữ liệu từ nhiều nguồn khác nhau.
  - **Pandas**: Cung cấp các hàm như *concat()* và *merge()* để kết hợp các DataFrame theo hàng, cột hoặc khóa.
  - **Dask**: Cung cấp các API tương tự như Pandas, bao gồm *concat()* và *merge()* để kết hợp các DataFrame.
  - **NumPy**: Cung cấp các hàm như *concatenate()* và *stack()* để kết hợp các mảng.

3. Khám phá dữ liệu

Phân tích thống kê

- Thực hiện các bài kiểm định thống kê để xác định mối quan hệ thống kê giữa các biến.
- **Kiểm định giả thuyết**
  - *statsmodels.stats.ttest_ind*: Kiểm định t cho hai mẫu độc lập.
  - *scipy.stats.ttest_rel*: Kiểm định t cho hai mẫu ghép đôi.
  - *scipy.stats.anova*: Phân tích phương sai (ANOVA).
  - *statsmodels.stats.chi2_contingency*: Kiểm định chi bình phương.
  - *scipy.stats.mannwhitneyu*: Kiểm định Mann-Whitney U.
- **Tương quan**
  - *scipy.stats.pearsonr*: Hệ số tương quan Pearson.
  - *scipy.stats.spearmanr*: Hệ số tương quan Spearman.

3. Khám phá dữ liệu

Phân tích thống kê

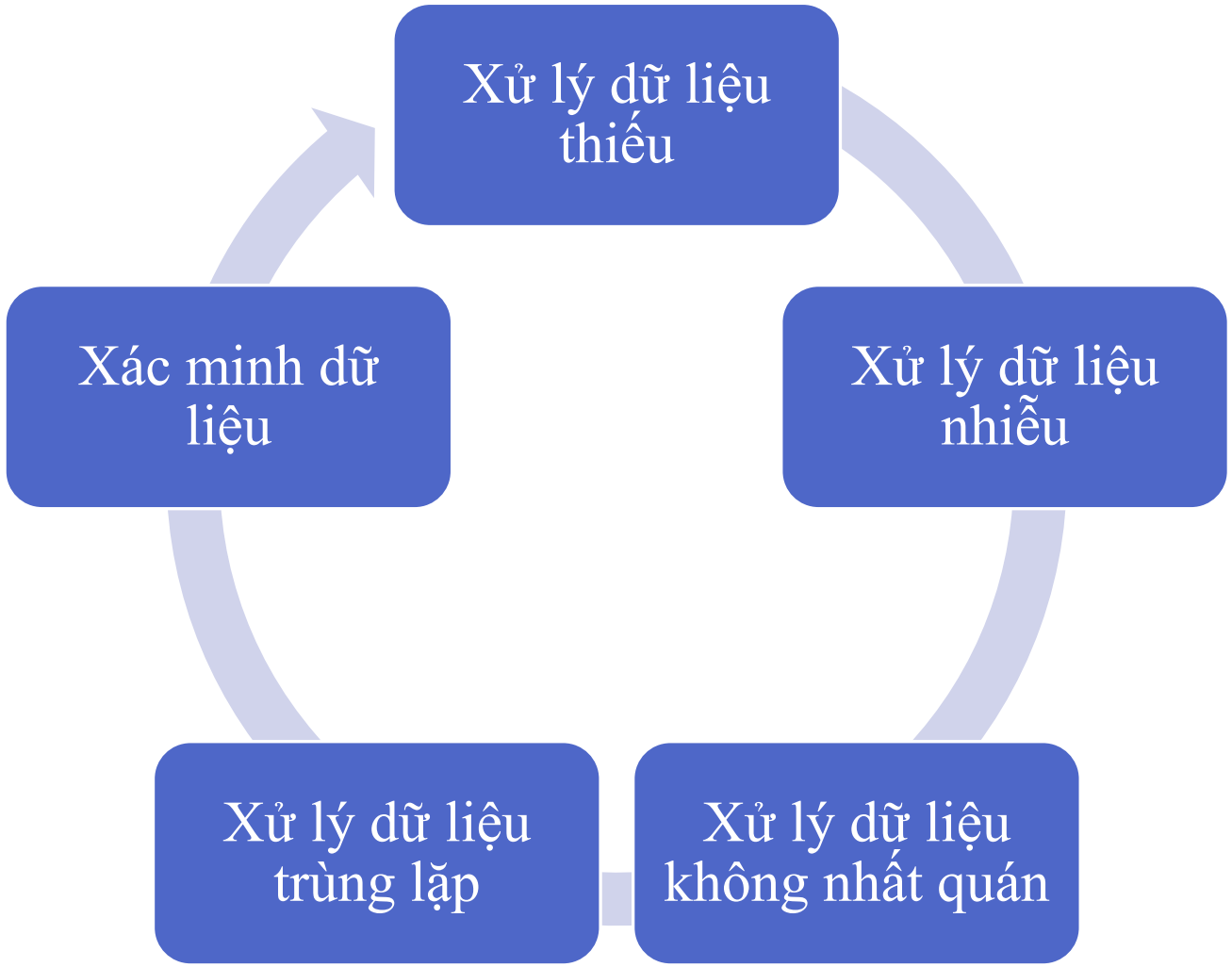
- Sử dụng các mô hình thống kê để dự đoán giá trị của biến mục tiêu.
- Áp dụng các kỹ thuật học máy để phân loại dữ liệu, dự đoán giá trị và khám phá các mẫu phức tạp.
- **Hồi quy**
  - *statsmodels.api.OLS*: Hồi quy tuyến tính.
  - *statsmodels.api.GLM*: Hồi quy tuyến tính tổng quát.
  - *statsmodels.api.Logit*: Hồi quy logistic.

3. Khám phá dữ liệu

Khai phá tri thức

- Xác định các mẫu, xu hướng và mối quan hệ ẩn trong dữ liệu.
- Sử dụng các kỹ thuật như khai phá luật kết hợp để khám phá các quy tắc ẩn trong dữ liệu.
- Tóm tắt và diễn giải các kết quả phân tích để dễ dàng hiểu và áp dụng.
- **Khai phá luật kết hợp (Association Rule Mining)**
  - *apriori*(transactions, min_support, min_confidence): Hàm khai phá luật kết hợp sử dụng thuật toán Apriori.
  - *fp_growth*(transactions, min_support): Hàm khai phá luật kết hợp sử dụng thuật toán FP-Growth.
  - *eclat*(transactions, min_support): Hàm khai phá luật kết hợp sử dụng thuật toán Eclat.

4. Làm sạch dữ liệu



Làm sạch dữ liệu

- Là một bước quan trọng.
- Giúp đảm bảo tính chính xác và hiệu quả cho các phân tích sau này.
- Cần hiểu rõ dữ liệu và mục đích phân tích để lựa chọn phương thức phù hợp.

4. Làm sạch dữ liệu

Xử lý dữ liệu thiếu

- Xóa bỏ: Xóa các điểm dữ liệu có quá nhiều giá trị thiếu.
- Thay thế: Thay thế giá trị thiếu bằng trung bình, trung vị, hoặc giá trị phổ biến nhất của cột dữ liệu.
- Dự đoán: Sử dụng các kỹ thuật học máy để dự đoán giá trị thiếu dựa trên các dữ liệu có sẵn.
- *Tương tự các bước xử lý dữ liệu trong quá trình "Khám phá dữ liệu". Quá trình xử lý dữ liệu có thể thực hiện nhiều lần ở các giai đoạn khác nhau trong vòng đời dữ liệu.*

4. Làm sạch dữ liệu

Xử lý dữ liệu nhiễu

- Cắt bớt: Loại bỏ các giá trị nằm ngoài phạm vi hợp lý.
- Làm mịn: Sử dụng các kỹ thuật thống kê để làm giảm nhiễu trong dữ liệu.
- Chuyển đổi: Chuyển đổi dữ liệu sang dạng khác để giảm nhiễu, ví dụ như chuyển đổi dữ liệu từ dạng số sang dạng logarit.
- *Tương tự các bước xử lý dữ liệu trong quá trình "Khám phá dữ liệu". Quá trình xử lý dữ liệu có thể thực hiện nhiều lần ở các giai đoạn khác nhau trong vòng đời dữ liệu.*

4. Làm sạch dữ liệu

Xử lý dữ liệu không nhất quán

- Chuẩn hóa: Chuyển đổi tất cả dữ liệu về cùng một đơn vị đo lường.
- Định dạng: Chuyển đổi dữ liệu sang định dạng phù hợp với các công cụ phân tích.
- Mã hóa: Chuyển đổi dữ liệu dạng văn bản sang dạng số để dễ dàng phân tích.
- *Tương tự các bước xử lý dữ liệu trong quá trình "Khám phá dữ liệu". Quá trình xử lý dữ liệu có thể thực hiện nhiều lần ở các giai đoạn khác nhau trong vòng đời dữ liệu.*

4. Làm sạch dữ liệu

Xử lý dữ liệu trùng lặp

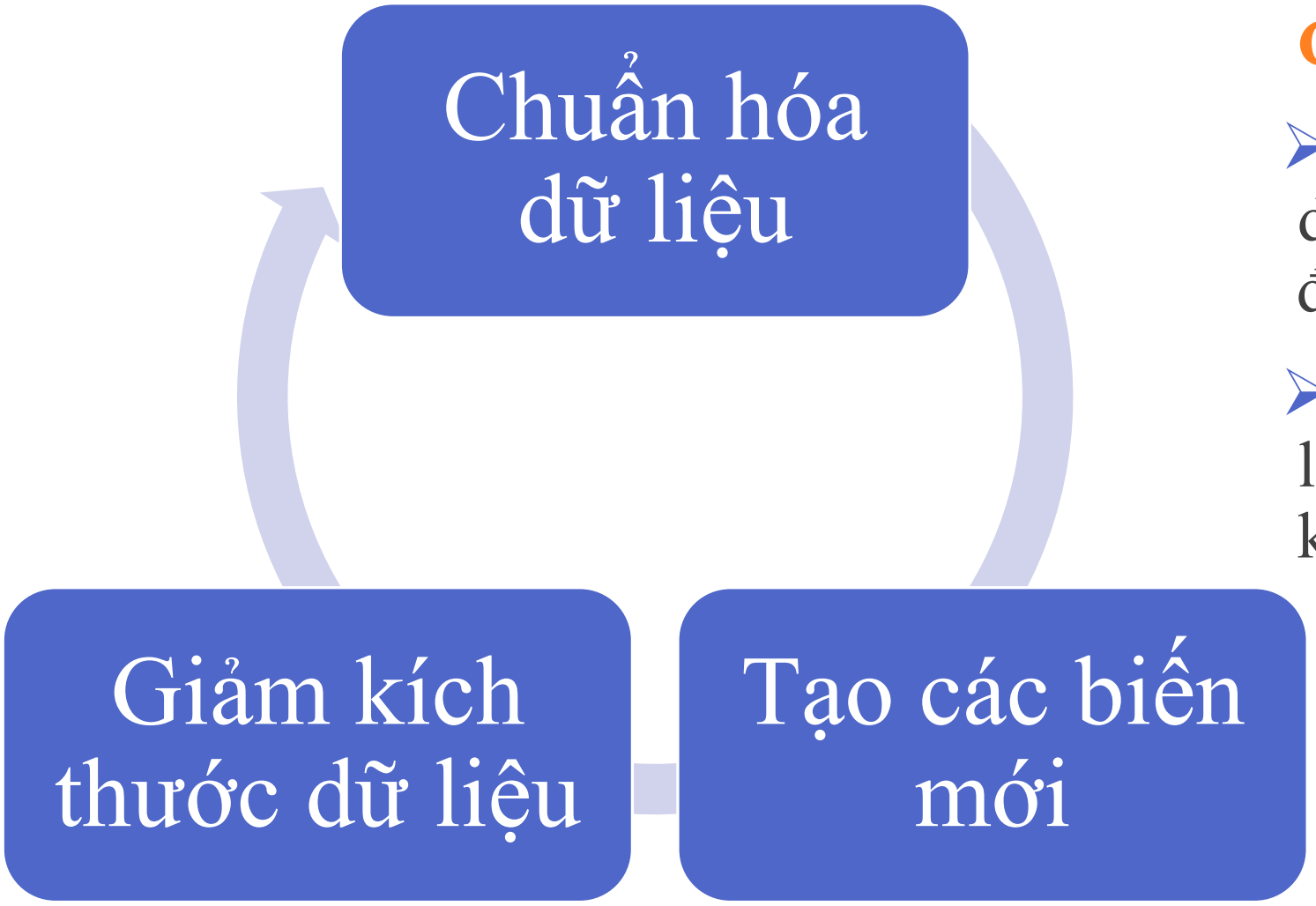
- Xác định: Sử dụng các kỹ thuật thống kê hoặc học máy để xác định các điểm dữ liệu trùng lặp.
- Loại bỏ: Xóa bỏ các điểm dữ liệu trùng lặp.
- **Pandas**
  - *DataFrame.duplicated*(): Xác định các bản ghi trùng lặp trong DataFrame.
  - *DataFrame.drop_duplicates*(): Xóa các bản ghi trùng lặp khỏi DataFrame.
- **NumPy**
  - *np.unique*(): Tìm các giá trị duy nhất trong một mảng NumPy.
- **Collections**
  - *Collections.Counter*(): Đếm số lần xuất hiện của các phần tử trong một tập hợp.

4. Làm sạch dữ liệu

Xác minh dữ liệu

- Kiểm tra tính chính xác: So sánh dữ liệu với các nguồn khác để đảm bảo tính chính xác.
- Kiểm tra tính hợp lý: Kiểm tra dữ liệu xem có hợp lý hay không.

5. Chuyển đổi dữ liệu



Chuyển đổi dữ liệu

- Giúp biến dữ liệu thô thành dạng thức phù hợp cho các mục đích phân tích và mô hình hóa.
- Giúp nâng cao chất lượng dữ liệu, tăng hiệu quả phân tích và khám phá thông tin mới

5. Chuyển đổi dữ liệu

Chuẩn hóa dữ liệu

- Loại bỏ dữ liệu nhiễu, thiếu sót: Xử lý các giá trị bất thường, thiếu thông tin để đảm bảo độ chính xác và tin cậy của dữ liệu.
- Chuyển đổi định dạng dữ liệu: Thống nhất định dạng dữ liệu (ví dụ: ngày tháng, đơn vị đo lường) để thuận tiện cho việc so sánh và phân tích.
- Mã hóa dữ liệu: Biến đổi các dữ liệu phi số (như tên, màu sắc) thành dạng số để sử dụng trong các thuật toán học máy.
- **Scikit-learn**
  - *sklearn.preprocessing.LabelEncoder*: Mã hóa các nhãn văn bản thành các số nguyên.
  - *sklearn.preprocessing.OrdinalEncoder*: Mã hóa các giá trị thứ tự (như xếp hạng) thành các số nguyên.
  - *sklearn.feature_extraction.DictVectorizer*: Chuyển đổi các từ điển sang ma trận thưa thớt với các giá trị nhị phân.

5. Chuyển đổi dữ liệu

Tạo các biến mới

- Kết hợp các biến: Tạo ra các biến mới từ các biến hiện có để khai thác thêm thông tin từ dữ liệu.
- Trích xuất đặc trưng: Sử dụng các kỹ thuật thống kê (*Phân tích thống kê trong khám phá dữ liệu*) và học máy để chọn ra các đặc trưng quan trọng nhất cho việc phân tích.
- **Pandas**
  - df['new_column'] = df['col1'] + df['col2']: Tạo cột mới bằng cách cộng hai cột hiện có.
  - df['new_column'] = df['col1'].fillna('missing'): Thay thế các giá trị thiếu trong cột col1 bằng chuỗi "missing".
  - df['new_column'] = df['col1'].str.upper(): Chuyển đổi tất cả các giá trị trong cột col1 sang chữ hoa.

5. Chuyển đổi dữ liệu

Giảm kích thước dữ liệu

- Lấy mẫu dữ liệu: Giảm số lượng dữ liệu để tăng tốc độ xử lý và giảm chi phí tính toán.
- Giảm chiều dữ liệu: Sử dụng các kỹ thuật như PCA (Phân tích thành phần chính) để giảm số lượng biến mà vẫn giữ được thông tin quan trọng.
- *Tương tự các bước trực quan hóa dữ liệu trong quá trình "Khám phá dữ liệu".*

6. Ghi nhãn dữ liệu

Ghi nhãn có giám sát

- Gắn nhãn thủ công bởi con người.
- Ví dụ: Bộ dữ liệu hình ảnh được dán nhãn như "mèo", "chó", "chim", v.v.

Ghi nhãn không giám sát

- Không được dán nhãn và mô hình học máy phải tự học cách phân loại dữ liệu.
- Ví dụ: Phân cụm (K-means, Gaussian Mixture Models, …), LOF (Local Outlier Factor) phát hiện điểm ngoại lai, v.v.

Ghi nhãn dữ liệu (hay dán nhãn dữ liệu)

➢Gắn nhãn cho các điểm dữ liệu với thông tin chính xác, giúp mô hình học máy hiểu và phân loại dữ liệu một cách hiệu quả.

6. Ghi nhãn dữ liệu

Một số công cụ và kỹ thuật có thể được sử dụng

- **Công cụ ghi nhãn dữ liệu**: Có nhiều công cụ ghi nhãn dữ liệu có sẵn để giúp tự động hóa quy trình và cải thiện hiệu quả.
- **Kỹ thuật lấy mẫu**: Kỹ thuật lấy mẫu có thể được sử dụng để giảm kích thước của bộ dữ liệu cần được dán nhãn.
- **Phương pháp học tập chủ động**: Phương pháp học tập chủ động có thể được sử dụng để chọn các điểm dữ liệu quan trọng nhất để dán nhãn, giúp giảm thiểu lượng công việc cần thiết.

7. Chia tập dữ liệu

Chia tập dữ liệu giúp đảm bảo tính chính xác và hiệu quả của các mô hình học máy được xây dựng.

- **Đánh giá hiệu quả mô hình**: Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra giúp đánh giá hiệu quả thực tế của mô hình trên dữ liệu chưa từng gặp.
- **Tránh hiện tượng quá khớp (overfitting)**: Quá khớp xảy ra khi mô hình học quá tốt tập huấn luyện, dẫn đến hiệu quả kém trên dữ liệu mới. Việc chia tập dữ liệu giúp mô hình tập trung vào việc học các đặc trưng chung của dữ liệu, hạn chế việc học các đặc trưng riêng lẻ chỉ có trong tập huấn luyện.
- **Tăng hiệu quả tính toán**: Chia tập dữ liệu thành các tập nhỏ hơn giúp giảm thời gian và tài nguyên cần thiết để huấn luyện mô hình.

7. Chia tập dữ liệu

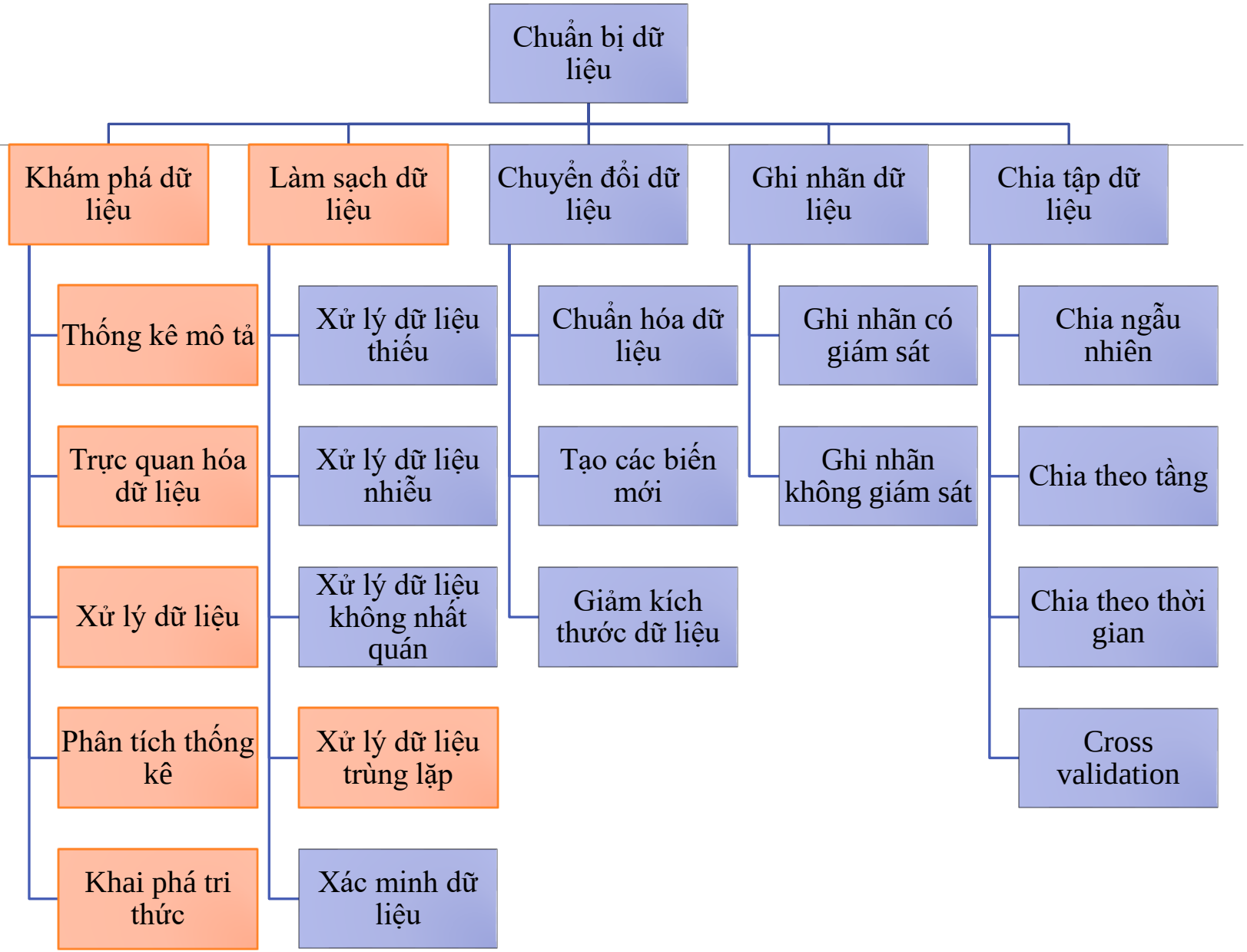
Các phương pháp chia tập dữ liệu phổ biến

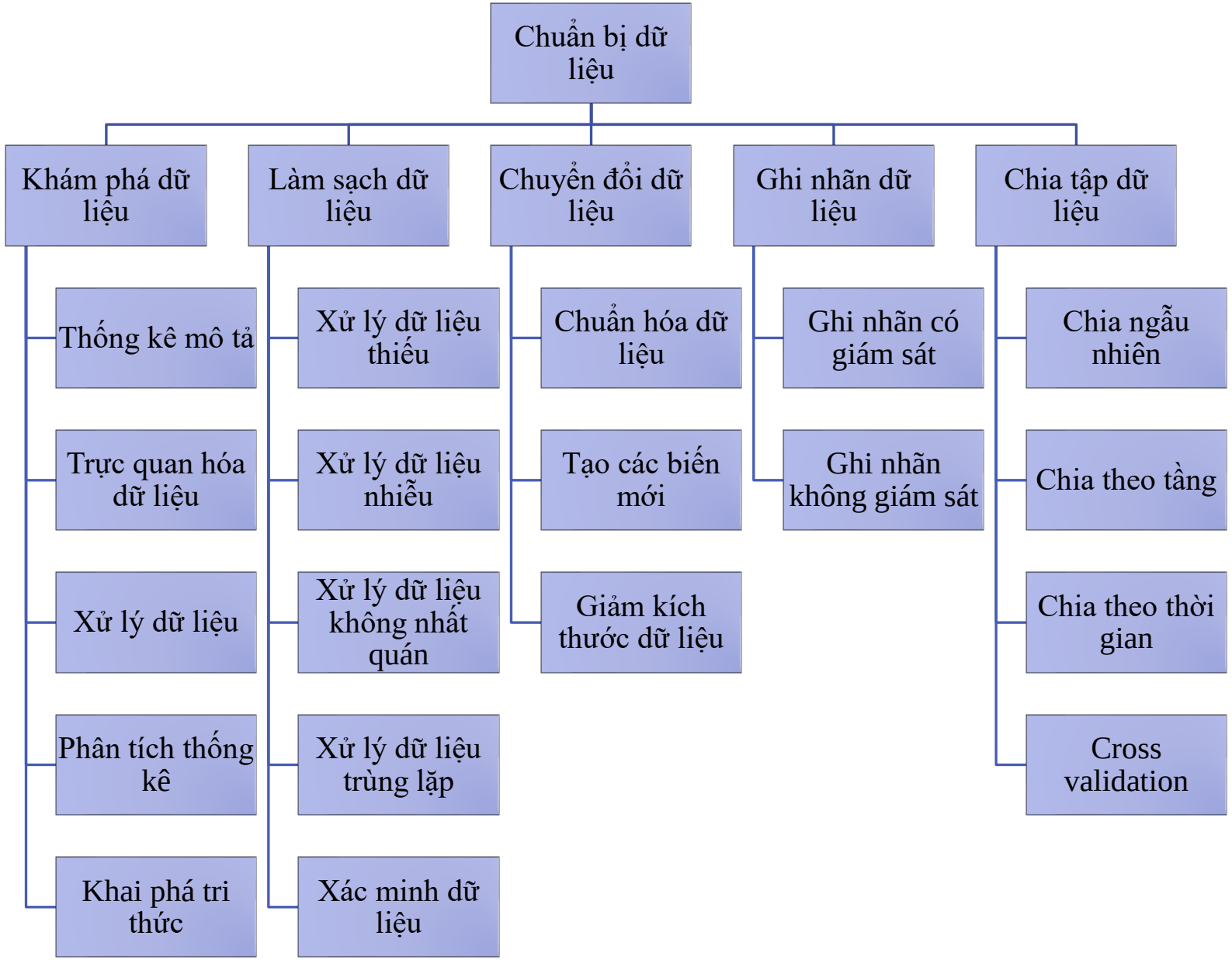
- **Chia ngẫu nhiên**: Đây là phương pháp đơn giản nhất, chia tập dữ liệu thành các tập con ngẫu nhiên.
- **Chia theo tầng** (stratified sampling): Phương pháp này chia tập dữ liệu thành các tập con đảm bảo tỷ lệ các lớp dữ liệu trong mỗi tập con tương tự như tỷ lệ trong tập dữ liệu gốc.
- **Chia theo thời gian** (time-based split): Phương pháp này chia tập dữ liệu theo thời gian, ví dụ như chia dữ liệu theo năm, tháng hoặc ngày.
- **Cross validation**: Chia tập dữ liệu thành nhiều phần (folds). Lặp lại quá trình huấn luyện và đánh giá mô hình nhiều lần, mỗi lần sử dụng một phần dữ liệu khác nhau để huấn luyện và phần còn lại để đánh giá.

8. Bài tập

Tìm hiểu công cụ và thư viện hỗ trợ - Python.

- Công dụng, cú pháp sử dụng, ví dụ demo cách áp dụng.
- Vận dụng vào tập dữ liệu khai thác của nhóm.




Chuẩn bị dữ liệu
Khám phá dữ liệu
Làm sạch dữ liệu
Chuyển đổi dữ liệu
Ghi nhãn dữ liệu
Chia tập dữ liệu
Thống kê mô tả
Trực quan hóa dữ liệu
Xử lý dữ liệu
Phân tích thống kê
Khai phá tri thức
Xử lý dữ liệu thiếu
Xử lý dữ liệu nhiễu
Xử lý dữ liệu không nhất quán
Xử lý dữ liệu trùng lặp
Xác minh dữ liệu
Chuẩn hóa dữ liệu
Tạo các biến mới
Giảm kích thước dữ liệu
Ghi nhãn có giám sát
Ghi nhãn không giám sát
Chia ngẫu nhiên
Chia theo tầng
Chia theo thời gian
Cross validation

9. Tổng kết

Question & Answer