

ĐỀ CƯƠNG ÔN TẬP

KHAI PHÁ DỮ LIỆU

Dành cho sinh viên đại học khối kinh tế, tài chính, công nghệ tài chính

Data Mining — a bilingual study guide

Dữ liệu và tiền xử lý · Luật kết hợp · Cây quyết định, Naive Bayes

kNN, hồi quy, mạng nơ-ron · Đánh giá mô hình

Phân cụm · Phát hiện bất thường, khai phá văn bản · Pháp luật về dữ liệu

Bài tập tính tay có lời giải · Tự luận có dàn ý · Trắc nghiệm có đáp án · Mã Python chạy thật

Song ngữ Việt – Anh · Vietnamese – English

Phiên bản 1.0 — Tháng 10 năm 2026

<https://globalmarket.vin>

M U C L U C

CONTENTS

Chương 1. Tổng quan về khai phá dữ liệu — Chapter 1. Introduction to data mining ..	6
1.1. Dữ liệu, thông tin và tri thức — 1.1. Data, information and knowledge	6
1.2. Khai phá dữ liệu và quá trình KDD — 1.2. Data mining and the KDD process	10
1.3. CRISP-DM và dự án chấm điểm tín dụng của Phú Sa — 1.3. CRISP-DM and Phú Sa's credit scoring project	13
1.4. Nhiệm vụ mô tả, nhiệm vụ dự đoán và các bài toán khai phá — 1.4. Descriptive and predictive tasks and the mining problems	16
1.5. Học có giám sát, không giám sát và bán giám sát — 1.5. Supervised, unsupervised and semi-supervised learning.....	19
1.6. Các loại dữ liệu được khai phá — 1.6. Types of data that are mined	23
1.7. Kho dữ liệu, OLAP và sự khác biệt với khai phá dữ liệu — 1.7. Data warehouse, OLAP and how they differ from data mining	26
1.8. Ứng dụng trong tài chính và công cụ khai phá dữ liệu — 1.8. Applications in finance and data mining tools	28
1.9. Quy trình dự án, vai trò nhân sự và thách thức — 1.9. Project process, team roles and challenges	31
1.10. Khung pháp lý tóm lược — 1.10. A short legal framework.....	33
1.11. Bảng thuật ngữ — 1.11. Glossary	35
1.12. Bài tập vận dụng có lời giải — Applied exercises with solutions	38
1.13. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines	40
1.14. Câu hỏi trắc nghiệm — Multiple-choice questions	42
Chương 2. Dữ liệu, thống kê mô tả và độ đo tương tự — Chapter 2. Data, descriptive statistics and similarity measures.....	44
2.1. Đối tượng, thuộc tính và kiểu thuộc tính — 2.1. Objects, attributes and attribute types	44
2.2. Đo xu hướng trung tâm — 2.2. Measures of central tendency	48
2.3. Đo độ phân tán và năm số tóm tắt — 2.3. Measures of dispersion and the five-number summary	51
2.4. Hình hộp, histogram và độ lệch — 2.4. Boxplots, histograms and skewness	55
2.5. Tương quan Pearson và biểu đồ phân tán — 2.5. Pearson correlation and the scatter plot.....	57
2.6. Ma trận dữ liệu, ma trận khoảng cách và khoảng cách Minkowski — 2.6. Data matrix, dissimilarity matrix and Minkowski distance.....	59
2.7. Độ tương tự của thuộc tính nhị phân: SMC và Jaccard — 2.7. Similarity of binary attributes: SMC and Jaccard	65
2.8. Thuộc tính định danh, thứ bậc và độ tương tự cosine — 2.8. Nominal and ordinal attributes and cosine similarity	67
2.9. Thuộc tính hỗn hợp và khoảng cách Gower — 2.9. Mixed attributes and the Gower distance.....	69

2.10. Bảng thuật ngữ — 2.10. Glossary	72
2.11. Bài tập vận dụng có lời giải — Applied exercises with solutions	74
2.12. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines	77
2.13. Câu hỏi trắc nghiệm — Multiple-choice questions	79
Chương 3. Tiền xử lý dữ liệu — Chapter 3. Data preprocessing	81
3.1. Chất lượng dữ liệu và các bước tiền xử lý — 3.1. Data quality and the preprocessing steps	81
3.2. Làm sạch: giá trị thiếu — 3.2. Cleaning: missing values	84
3.3. Làm sạch: nhiễu, chia giỏ và làm trơn — 3.3. Cleaning: noise, binning and smoothing	86
3.4. Ngoại lai: phát hiện theo IQR và z-score — 3.4. Outliers: detection by IQR and z-score	89
3.5. Tích hợp dữ liệu — 3.5. Data integration	92
3.6. Biến đổi: chuẩn hóa dữ liệu — 3.6. Transformation: normalisation	94
3.7. Biến đổi: mã hóa, rời rạc hóa và thuộc tính mới — 3.7. Transformation: encoding, discretisation and new attributes	97
3.8. Thu giảm số chiều: chọn thuộc tính và PCA — 3.8. Dimensionality reduction: attribute selection and PCA	100
3.9. Lấy mẫu và mất cân bằng lớp — 3.9. Sampling and class imbalance	104
3.10. Rò rỉ dữ liệu và thứ tự đúng của các bước — 3.10. Data leakage and the correct order of steps	106
3.11. Bảng thuật ngữ — 3.11. Glossary	108
3.12. Bài tập vận dụng có lời giải — Applied exercises with solutions	109
3.13. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines	111
3.14. Câu hỏi trắc nghiệm — Multiple-choice questions	113
Chương 4. Khai phá luật kết hợp — Chapter 4. Association rule mining	115
4.1. Bài toán phân tích giỏ hàng và bán chéo sản phẩm tài chính — 4.1. Market basket analysis and cross-selling of financial products	115
4.2. Tập mục, độ hỗ trợ, độ tin cậy, lift và các độ đo khác — 4.2. Itemsets, support, confidence, lift and other measures	118
4.3. Thuật toán Apriori: tính chất đóng xuống, ghép và tỉa — 4.3. The Apriori algorithm: downward closure, join and prune	121
4.4. Bài mẫu 1: chạy tay Apriori trên giao dịch sản phẩm của Phú Sa — 4.4. Worked example 1: Apriori by hand on the Phú Sa product transactions	124
4.5. Sinh luật từ tập phổ biến và lọc theo độ tin cậy — 4.5. Generating rules from frequent itemsets and filtering by confidence	127
4.6. Luật mạnh nhưng không thú vị: đánh giá bằng lift, leverage và conviction — 4.6. Strong but uninteresting rules: evaluation with lift, leverage and conviction	130
4.7. Tập phổ biến đóng và tập phổ biến cực đại — 4.7. Closed and maximal frequent itemsets	132
4.8. Thuật toán FP-growth và cây FP-tree — 4.8. The FP-growth algorithm and the FP-tree	134

4.9. So sánh Apriori và FP-growth; luật nhiều mức và luật định lượng — 4.9. <i>Apriori compared with FP-growth; multilevel and quantitative rules</i>	139
4.10. Mẫu tuần tự, ứng dụng đề xuất sản phẩm trên ứng dụng Phú Sa và giới hạn — 4.10. <i>Sequential patterns, product recommendation in the Phú Sa app and limitations</i> ..	142
4.11. Bảng thuật ngữ — 4.11. <i>Glossary</i>	144
4.12. Bài tập vận dụng có lời giải — <i>Applied exercises with solutions</i>	146
4.13. Câu hỏi tự luận và dàn ý trả lời — <i>Essay questions and answer outlines</i>	149
4.14. Câu hỏi trắc nghiệm — <i>Multiple-choice questions</i>	151

Chương 5. Phân lớp: cây quyết định và Naive Bayes — Chapter 5. Classification: decision trees and Naive Bayes 153

5.1. Bài toán phân lớp và dữ liệu hồ sơ vay — 5.1. <i>The classification problem and the loan-application data</i>	153
5.2. Cây quyết định: cấu trúc và quy nạp từ trên xuống — 5.2. <i>Decision trees: structure and top-down induction</i>	156
5.3. Entropy và information gain: tiêu chí của ID3 — 5.3. <i>Entropy and information gain: the ID3 criterion</i>	158
5.4. Gain ratio, chỉ số Gini và phép tách nhị phân — 5.4. <i>Gain ratio, the Gini index and binary splits</i>	160
5.5. Dựng cây ID3 hoàn chỉnh và rút luật IF–THEN — 5.5. <i>Building the full ID3 tree and extracting IF–THEN rules</i>	165
5.6. Thuộc tính liên tục và ngưỡng tách — 5.6. <i>Continuous attributes and split thresholds</i>	169
5.7. Quá khớp, tia cây, ưu điểm và nhược điểm — 5.7. <i>Overfitting, pruning, strengths and weaknesses</i>	171
5.8. Naive Bayes: định lý Bayes, làm trơn Laplace và log-xác suất — 5.8. <i>Naive Bayes: Bayes' theorem, Laplace smoothing and log-probabilities</i>	174
5.9. Gaussian Naive Bayes và so sánh với cây quyết định — 5.9. <i>Gaussian Naive Bayes and comparison with decision trees</i>	178
5.10. Bảng thuật ngữ — 5.10. <i>Glossary</i>	182
5.11. Bài tập vận dụng có lời giải — <i>Applied exercises with solutions</i>	184
5.12. Câu hỏi tự luận và dàn ý trả lời — <i>Essay questions and answer outlines</i>	187
5.13. Câu hỏi trắc nghiệm — <i>Multiple-choice questions</i>	189

Chương 6. Phân lớp nâng cao và mô hình dự báo — Chapter 6. Advanced classification and predictive models 191

6.1. Hai bài toán dự đoán và bộ hồ sơ thử nghiệm — 6.1. <i>Two prediction problems and the test applications</i>	191
6.2. k láng giềng gần nhất (kNN) — 6.2. <i>k-nearest neighbours (kNN)</i>	193
6.3. Hồi quy tuyến tính dự báo dư nợ — 6.3. <i>Linear regression to forecast outstanding loans</i>	198
6.4. Hồi quy logistic — 6.4. <i>Logistic regression</i>	202
6.5. Perceptron, mạng nơ-ron và học sâu — 6.5. <i>The perceptron, neural networks and deep learning</i>	204
6.6. Máy véc-tơ hỗ trợ (SVM) — 6.6. <i>Support vector machines (SVM)</i>	209

6.7. Học kết hợp: bagging, rừng ngẫu nhiên, boosting — 6.7. Ensemble learning: bagging, random forests, boosting	212
6.8. So sánh các bộ phân lớp và chọn mô hình — 6.8. Comparing classifiers and choosing a model	216
6.9. Bảng thuật ngữ — 6.9. Glossary	218
6.10. Bài tập vận dụng có lời giải — Applied exercises with solutions	221
6.11. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines	223
6.12. Câu hỏi trắc nghiệm — Multiple-choice questions	225

Chương 7. Đánh giá và lựa chọn mô hình — Chapter 7. Model evaluation and selection

7.1. Ma trận nhầm lẫn và bộ dữ liệu điểm dự báo — 7.1. The confusion matrix and the predicted-score dataset	227
7.2. Accuracy, precision, recall, specificity, F1 và F β — 7.2. Accuracy, precision, recall, specificity, F1 and F β	229
7.3. Ngưỡng quyết định và đánh đổi precision – recall — 7.3. The decision threshold and the precision–recall trade-off	232
7.4. Đường ROC, AUC và Gini — 7.4. The ROC curve, AUC and the Gini coefficient	235
7.5. Đường precision – recall, gain và lift — 7.5. The precision–recall curve, gain and lift	238
7.6. Ma trận chi phí và chọn ngưỡng — 7.6. The cost matrix and threshold selection	240
7.7. Ước lượng sai số: holdout, k-fold, leave-one-out, bootstrap — 7.7. Estimating error: holdout, k-fold, leave-one-out, bootstrap	243
7.8. Quá khớp, chưa khớp và bias – variance — 7.8. Overfitting, underfitting and bias–variance	247
7.9. So sánh hai mô hình bằng khoảng tin cậy — 7.9. Comparing two models with a confidence interval	250
7.10. Chỉ số hồi quy và đánh giá trong chấm điểm tín dụng — 7.10. Regression measures and evaluation in credit scoring	255
7.11. Bảng thuật ngữ — 7.11. Glossary	257
7.12. Bài tập vận dụng có lời giải — Applied exercises with solutions	259
7.13. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines	261
7.14. Câu hỏi trắc nghiệm — Multiple-choice questions	263

Chương 8. Phân tích cụm — Chapter 8. Cluster analysis

8.1. Khái niệm, yêu cầu và bài toán phân cụm — 8.1. Concept, requirements and the clustering task	265
8.2. Phân loại các phương pháp phân cụm — 8.2. Families of clustering methods	267
8.3. Chuẩn hóa dữ liệu trước khi phân cụm — 8.3. Scaling the data before clustering	269
8.4. Phân cụm phân hoạch: thuật toán k-means — 8.4. Partitioning: the k-means algorithm	270
8.5. Chọn số cụm k và đánh giá chất lượng cụm — 8.5. Choosing k and judging cluster quality	275
8.6. Nhược điểm của k-means, k-means++ và k-medoids — 8.6. Weaknesses of k-means, k-means++ and k-medoids	279

8.7. Phân cụm phân cấp — 8.7. Hierarchical clustering	283
8.8. Phân cụm theo mật độ: DBSCAN — 8.8. Density-based clustering: DBSCAN.....	287
8.9. Phân khúc khách hàng bằng RFM — 8.9. Customer segmentation with RFM.....	291
8.10. Bảng thuật ngữ — 8.10. Glossary	294
8.11. Bài tập vận dụng có lời giải — Applied exercises with solutions	296
8.12. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines	298
8.13. Câu hỏi trắc nghiệm — Multiple-choice questions	300
Chương 9. Phát hiện bất thường, khai phá văn bản và các vấn đề pháp lý – đạo đức — Chapter 9. Anomaly detection, text mining, and legal and ethical issues	303
9.1. Bất thường và ngoại lai — 9.1. Anomalies and outliers	303
9.2. Phương pháp thống kê: z-score, IQR và MAD — 9.2. Statistical methods: z-score, IQR and MAD.....	305
9.3. Phương pháp dựa khoảng cách, mật độ và mô hình — 9.3. Distance-, density- and model-based methods	309
9.4. Ứng dụng chống gian lận thẻ và phòng chống rửa tiền — 9.4. Applications to card fraud and anti-money-laundering	313
9.5. Chuỗi thời gian: thành phần, trung bình trượt và san mũ — 9.5. Time series: components, moving average and exponential smoothing.....	315
9.6. Khai phá văn bản: tách từ, TF-IDF và độ tương tự cosin — 9.6. Text mining: word segmentation, TF-IDF and cosine similarity	320
9.7. Dữ liệu lớn và mô hình MapReduce — 9.7. Big data and the MapReduce model...	325
9.8. Khung pháp lý về dữ liệu và bí mật thông tin khách hàng — 9.8. Legal framework for data and customer-information secrecy.....	327
9.9. Trí tuệ nhân tạo và chấm điểm tự động — 9.9. Artificial intelligence and automated scoring.....	331
9.10. Đạo đức: thiên lệch, giải thích được, ẩn danh hóa và k-anonymity — 9.10. Ethics: bias, explainability, anonymisation and k-anonymity.....	334
9.11. Bảng thuật ngữ — 9.11. Glossary	339
9.12. Bài tập vận dụng có lời giải — Applied exercises with solutions	341
9.13. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines	344
9.14. Câu hỏi trắc nghiệm — Multiple-choice questions	346
Phụ lục A. Đáp án câu hỏi trắc nghiệm — Appendix A. Answer key.....	349
Phụ lục B. Công thức khai phá dữ liệu — Appendix B. Data mining formulas.....	354
Phụ lục C. Thuật ngữ Việt – Anh — Appendix C. Vietnamese–English glossary.....	357
Phụ lục D. Văn bản pháp luật về dữ liệu và trí tuệ nhân tạo — Appendix D. Vietnamese legislation on data and artificial intelligence	365
Tài liệu tham khảo — References.....	366

Chương 1. Tổng quan về khai phá dữ liệu — Chapter 1. Introduction to data mining

Mục tiêu chương — Chapter objectives

- Phân biệt dữ liệu, thông tin và tri thức; phân biệt KDD với khai phá dữ liệu và nêu năm bước của quá trình.
Distinguish data, information and knowledge; distinguish KDD from data mining and state the five steps of the process.
- Mô tả sáu pha CRISP-DM và áp vào dự án chấm điểm tín dụng của Công ty Tài chính TNHH Phú Sa.
Describe the six CRISP-DM phases and apply them to the credit scoring project of Phú Sa Finance Company Limited.
- Xếp một câu hỏi kinh doanh vào bài toán khai phá (phân lớp, hồi quy, phân cụm, luật kết hợp, bất thường, chuỗi, văn bản) và kiểu học tương ứng.
Place a business question under a mining problem (classification, regression, clustering, association rules, anomalies, sequences, text) and the matching learning type.
- Đọc bảng dữ liệu nhỏ, đếm, tính tỷ lệ và chấm một quy tắc phân lớp đơn giản bằng tay và bằng mã Python.
Read a small data table, count, compute rates and score a simple classification rule by hand and in Python code.
- Phân biệt kho dữ liệu, OLAP với khai phá dữ liệu; nêu thách thức của dự án và các quy định về dữ liệu cá nhân, bí mật thông tin khách hàng.
Tell a data warehouse and OLAP from data mining; state the challenges of a project and the rules on personal data and customer information confidentiality.

1.1. Dữ liệu, thông tin và tri thức — 1.1. Data, information and knowledge

Dữ liệu (data) là giá trị ghi lại về một đối tượng hay một sự kiện, chưa gắn với câu hỏi nào. Dòng "KH03, 34, Giáo viên, Vàng" trong bảng khách hàng của Công ty Tài chính TNHH Phú Sa (TP. Cần Thơ) là dữ liệu. Thông tin (information) là dữ liệu đã được tổ chức và đặt vào ngữ cảnh để trả lời một câu hỏi cụ thể, chẳng hạn "12 khách hàng, trong đó 7 người có nhà". Tri thức (knowledge) là mẫu hay quy luật rút ra từ nhiều thông tin, dùng được để quyết định cho trường hợp chưa gặp, chẳng hạn "hồ sơ có lịch sử tín dụng Tốt hiếm khi thành nợ xấu". Ba cấp này là phần dưới của mô hình DIKW (data, information, knowledge, wisdom). Khai phá dữ liệu dùng thuật toán để đi từ cấp thứ nhất lên cấp thứ ba.

Data is a recorded value about an object or an event, not yet tied to any question. The row "KH03, 34, Teacher, Gold" in the customer table of Phú Sa Finance Company Limited (Cần Thơ City) is data. Information is data that has been organised and placed in context to answer a specific question, for example "12 customers, 7 of whom own a house". Knowledge is a pattern or rule drawn from many pieces of information and usable for a decision on a case not yet seen, for example "applications with a Good credit history rarely become bad debt". The three levels

form the lower part of the DIKW model (data, information, knowledge, wisdom). Data mining uses algorithms to climb from the first level to the third.

Cấp Level	Đặc điểm Feature	Ví dụ ở Phú Sa Example at Phú Sa
Dữ liệu Data	Giá trị thô, chưa có ngữ cảnh, đọc một mình chưa trả lời được câu hỏi nào <i>Raw value without context; read alone it answers no question</i>	Dòng của KH07: 45 tuổi, kinh doanh tự do, thiếu thu nhập <i>The row of KH07: aged 45, self-employed, income missing</i>
Thông tin Information	Dữ liệu đã đếm, tổng hợp, so sánh để trả lời câu hỏi "bao nhiêu, ở đâu, khi nào" <i>Data counted, summarised and compared to answer "how many, where, when"</i>	7 trong 12 khách có nhà (58,3%) <i>7 of 12 customers own a house (58.3%)</i>
Tri thức Knowledge	Mẫu, quy luật hoặc mô hình khái quát được cho dữ liệu mới, dùng để quyết định <i>A pattern, rule or model that generalises to new data and supports a decision</i>	Lịch sử tín dụng Xấu đi cùng tỷ lệ nợ xấu 80%, lịch sử Tốt đi cùng 0% <i>A Bad credit history goes with an 80% bad-debt rate, a Good one with 0%</i>

Mã sau đọc 12 dòng bảng khách hàng (dữ kiện đề bài rút gọn còn năm cột) và đếm. Mỗi dòng in ra là một mẫu thông tin.

The following code reads the 12 rows of the customer table (a problem statement reduced to five columns) and counts. Each printed line is one piece of information.

```
from collections import Counter

khach = [ # (mã, tuổi, thu nhập triệu/tháng, nghề, có nhà)
    ("KH01", 25, 9, "Công nhân", 0),
    ("KH02", 29, 12, "Nhân viên văn phòng", 0),
    ("KH03", 34, 15, "Giáo viên", 1),
    ("KH04", 41, 22, "Kinh doanh tự do", 1),
    ("KH05", 23, 8, "Công nhân", 0),
    ("KH06", 38, 18, "Nhân viên văn phòng", 1),
    ("KH07", 45, None, "Kinh doanh tự do", 1),
    ("KH08", 31, 14, "Giáo viên", 0),
    ("KH09", 27, 11, "Nhân viên văn phòng", 0),
    ("KH10", 36, 16, "Công nhân", 1),
    ("KH11", 50, 20, "Giáo viên", 1),
    ("KH12", 47, 120, "Kinh doanh tự do", 1),
]
n = len(khach)
print("Số khách:", n)
print("Thiếu thu nhập:", [k[0] for k in khach if k[2] is None])
for nghe, c in Counter(k[3] for k in khach).items():
    print(f" {nghe}: {c} khách")
co_nha = sum(k[4] for k in khach)
print(f"Có nhà: {co_nha}/{n} = {co_nha / n:.1%}")
tuoi = Counter("<30" if k[1] < 30 else "30-39" if k[1] < 40 else ">=40"
               for k in khach)
print("Nhóm tuổi:", dict(tuoi))
```

Kết quả chạy / Output

```
Số khách: 12
Thiếu thu nhập: ['KH07']
Công nhân: 3 khách
Nhân viên văn phòng: 3 khách
Giáo viên: 3 khách
Kinh doanh tự do: 3 khách
Có nhà: 7/12 = 58.3%
Nhóm tuổi: {'<30': 4, '30-39': 4, '>=40': 4}
```

Bảng có 12 khách, một ô thiếu (thu nhập của KH07), mỗi nghề 3 khách, 7 trên 12 khách (58,3%) có nhà, mỗi nhóm tuổi 4 khách. Các con số mô tả bảng nhưng chưa giải thích vì sao khách này có nhà còn khách kia thì không; đó là việc của các bước sau.

The table has 12 customers, one missing cell (the income of KH07), 3 customers in each occupation, 7 of 12 customers (58.3%) owning a house, and 4 customers in each age group. The figures describe the table but do not explain why one customer owns a house and another does not; that is the job of the later steps.

Bài mẫu 1. Từ thông tin đến quy tắc — *Worked example 1. From information to a rule*

Đề bài: bảng 14 hồ sơ vay gồm lịch sử tín dụng (Tốt, Xấu, Chưa có) và nhãn nợ xấu (5 hồ sơ có nợ xấu). Tính tỷ lệ nợ xấu từng nhóm, đề xuất quy tắc phân lớp dùng một thuộc tính, tính độ chính xác và so với cách luôn đoán lớp đông nhất.

Problem: a table of 14 loan applications with credit history (Good, Bad, None) and a bad-debt label (5 applications with bad debt). Compute the bad-debt rate of each group, propose a classification rule using one attribute, compute its accuracy and compare it with always guessing the largest class.

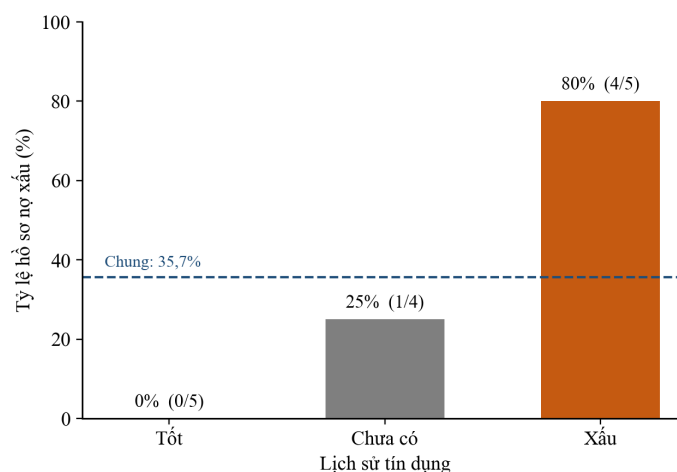
```
# (lịch sử tín dụng, nợ xấu: 1 = có) của 14 hồ sơ HS01 đến HS14
ho_so = [("Xấu", 1), ("Chưa có", 1), ("Xấu", 1), ("Xấu", 1),
          ("Xấu", 1), ("Xấu", 0), ("Tốt", 0), ("Tốt", 0),
          ("Tốt", 0), ("Tốt", 0), ("Tốt", 0), ("Chưa có", 0),
          ("Chưa có", 0), ("Chưa có", 0)]
nhom = {}
for ls, xau in ho_so:
    n, x = nhom.get(ls, (0, 0))
    nhom[ls] = (n + 1, x + xau)
for ls, (n, x) in nhom.items():
    print(f"{ls:8s} hồ sơ: {n}, nợ xấu: {x}, tỷ lệ {x / n:.0%}")
tong = sum(x for _, x in nhom.values())
print(f"Trung bình {tong}/{len(ho_so)} = {tong / len(ho_so):.1%}")
dung = sum(max(x, n - x) for n, x in nhom.values())
print(f"Đoán theo đa số của nhóm: {dung}/{len(ho_so)}")
mac_dinh = max(tong, len(ho_so) - tong)
print(f"Luôn đoán lớp đông nhất: {mac_dinh}/{len(ho_so)}")
```

Kết quả chạy / Output

```
Xấu      hồ sơ: 5, nợ xấu: 4, tỷ lệ 80%
Chưa có   hồ sơ: 4, nợ xấu: 1, tỷ lệ 25%
Tốt       hồ sơ: 5, nợ xấu: 0, tỷ lệ 0%
Chung     5/14 = 35.7%
Đoán theo đa số của nhóm: 12/14
Luôn đoán lớp đông nhất: 9/14
```

Nhóm lịch sử <i>History group</i>	Số hồ sơ <i>Applications</i>	Nợ xấu <i>Bad debt</i>	Tỷ lệ <i>Rate</i>	Lớp đoán (đa số) <i>Predicted class (majority)</i>	Đ đoán đúng <i>Correct</i>
Xấu <i>Bad</i>	5	4	80%	Nợ xấu <i>Bad debt</i>	4
Chưa có <i>None</i>	4	1	25%	Không nợ xấu <i>No bad debt</i>	3
Tốt <i>Good</i>	5	0	0%	Không nợ xấu <i>No bad debt</i>	5
Cộng <i>Total</i>	14	5	35,7%		12

- Đếm số hồ sơ và số hồ sơ nợ xấu của từng nhóm: Xấu 5 và 4, Chưa có 4 và 1, Tốt 5 và 0. Tổng 14 hồ sơ, 5 nợ xấu.
Count the applications and the bad-debt applications in each group: Bad 5 and 4, None 4 and 1, Good 5 and 0. In total 14 applications, 5 with bad debt.
- Tính tỷ lệ từng nhóm: $4/5 = 80\%$, $1/4 = 25\%$, $0/5 = 0\%$. Tỷ lệ chung $5/14 = 35,7\%$.
Compute each group's rate: $4/5 = 80\%$, $1/4 = 25\%$, $0/5 = 0\%$. The overall rate is $5/14 = 35.7\%$.
- Lập quy tắc: mỗi nhóm đoán theo lớp chiếm đa số. Nhóm Xấu đoán nợ xấu; hai nhóm còn lại đoán không nợ xấu.
Form the rule: each group is predicted as its majority class. The Bad group is predicted as bad debt; the other two groups as no bad debt.
- Số hồ sơ đoán đúng là $4 + 3 + 5 = 12$ trên 14, độ chính xác $85,7\%$. Cách luôn đoán "không nợ xấu" (9 hồ sơ) đúng $9/14 = 64,3\%$.
The number of correct predictions is $4 + 3 + 5 = 12$ out of 14, an accuracy of 85.7% . Always guessing "no bad debt" (9 applications) is right $9/14 = 64.3\%$ of the time.



Hình 1.1. Tỷ lệ hồ sơ nợ xấu theo lịch sử tín dụng trong 14 hồ sơ vay; đường nét đứt là tỷ lệ chung

Figure 1.1. Share of applications with bad debt by credit history among 14 loan applications; the dashed line is the overall share

Quy tắc trên hơn cách đoán mặc định 21,4 điểm phần trăm (85,7% so với 64,3%). Mẫu chỉ có 14 hồ sơ nên kết luận này mới là giả thuyết; Chương 7 trình bày cách kiểm trên dữ liệu chưa dùng để lập quy tắc.

The rule beats the default guess by 21.4 percentage points (85.7% against 64.3%). With only 14 applications this is still a hypothesis; Chapter 7 shows how to test it on data not used to build the rule.

PHẢI THUỘC / MEMORISE — Dữ liệu là giá trị thô. Thông tin là dữ liệu đã tổng hợp để trả lời một câu hỏi. Tri thức là mẫu hay quy luật dùng được cho dữ liệu mới và phải được kiểm trên dữ liệu chưa dùng khi lập nó.

Data is a raw value. Information is data summarised to answer a question. Knowledge is a pattern or rule usable on new data, and it has to be tested on data not used to build it.

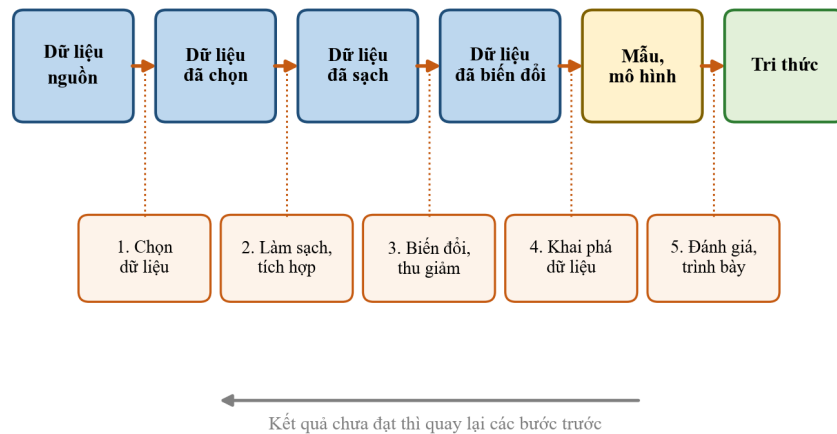
LỖI HAY GẶP / COMMON MISTAKE — Gọi mọi con số in ra là tri thức. "Có 12 khách" và "tổng dư nợ 415 triệu đồng" là thông tin; chỉ khi con số chuyển thành quy tắc dùng cho khách mới (ví dụ lịch sử Tốt thì ít rủi ro) và qua kiểm tra thì mới là tri thức.

Calling every printed number knowledge. "There are 12 customers" and "total debt 415 million dong" are information; only when a figure becomes a rule for new customers (for example a Good history means low risk) and passes a test is it knowledge.

1.2. Khai phá dữ liệu và quá trình KDD — 1.2. Data mining and the KDD process

Khai phá dữ liệu (data mining) là quá trình dùng thuật toán để tìm mẫu và mô hình có ích trong tập dữ liệu lớn, thay cho việc người phân tích tự nêu từng giả thuyết rồi kiểm từng cái. Tên gọi dễ gây hiểu nhầm: thứ được khai phá là tri thức nằm trong dữ liệu, giống như khai thác vàng từ quặng. Khám phá tri thức trong cơ sở dữ liệu (knowledge discovery in databases, KDD) là cả quá trình từ dữ liệu nguồn đến tri thức; khai phá dữ liệu là bước thứ tư trong năm bước. Nhiều tài liệu dùng hai tên như nhau, nhưng khi đề thi hỏi phân biệt thì khai phá dữ liệu chỉ là một bước của KDD.

Data mining is the process of using algorithms to find useful patterns and models in large data sets, instead of the analyst stating each hypothesis and testing them one by one. The name is misleading: what is mined is the knowledge lying in the data, as gold is mined from ore. Knowledge discovery in databases (KDD) is the whole process from source data to knowledge; data mining is the fourth of its five steps. Many texts use the two names interchangeably, but when an exam asks for the difference, data mining is only one step of KDD.



Hình 1.2. Quá trình KDD: sáu dạng dữ liệu nối nhau bằng năm bước, bước thứ tư là khai phá dữ liệu

Figure 1.2. The KDD process: six forms of data joined by five steps, the fourth being data mining

Bước Step	Việc làm What is done	Ví dụ ở Phú Sa Example at Phú Sa
1. Chọn dữ liệu <i>1. Select data</i>	Xác định bảng, cột, khoảng thời gian liên quan đến bài toán <i>Identify the tables, columns and period relevant to the problem</i>	Lấy thu nhập, nghề, hạng thẻ, lịch sử trả nợ; bỏ cột ghi chú tự do <i>Take income, occupation, card tier and repayment history; drop the free-text note</i>
2. Làm sạch, tích hợp <i>2. Clean, integrate</i>	Xử lý giá trị thiếu, nhiễu, trùng; ghép nhiều nguồn về một bảng <i>Handle missing values, noise and duplicates; merge several sources into one table</i>	Điền thu nhập thiếu của KH07; ghép bảng khách với bảng khoản vay theo mã khách <i>Fill the missing income of KH07; join the customer and loan tables on customer code</i>
3. Biến đổi, thu giảm <i>3. Transform, reduce</i>	Chuẩn hóa, rời rạc hóa, tạo thuộc tính mới, giảm số chiều <i>Normalise, discretise, derive new attributes, reduce dimensions</i>	Tính DTI (nợ trên thu nhập); chia thu nhập thành ba mức <i>Compute DTI (debt over income); split income into three levels</i>
4. Khai phá dữ liệu <i>4. Data mining</i>	Chạy thuật toán phân lớp, phân cụm, luật kết hợp để tìm mẫu <i>Run classification, clustering or association algorithms to find patterns</i>	Dựng mô hình đoán nợ xấu cho hồ sơ mới <i>Build a model that predicts bad debt for a new application</i>
5. Đánh giá, trình bày <i>5. Evaluate, present</i>	Loại mẫu tầm thường, đo độ chính xác, trình bày bằng bảng và hình để người dùng quyết định <i>Discard trivial patterns, measure accuracy, present with tables and figures for the user to decide</i>	Báo cáo cho khối tín dụng: ngưỡng điểm duyệt, tỷ lệ bỏ sót nợ xấu <i>Report to the credit division: the approval score cut-off and the share of missed bad debts</i>

Một mẫu tìm được chỉ có giá trị khi đạt bốn điều kiện: hợp lệ trên dữ liệu mới (valid), chưa biết trước (novel), có thể dùng vào việc cụ thể (potentially useful) và người dùng

hiểu được (understandable). Quy tắc "khách có khoản vay thì có dư nợ" hợp lệ nhưng tầm thường, nên bị loại ở bước 5.

A pattern found has value only if it meets four conditions: valid on new data, not known beforehand (novel), usable for a concrete task (potentially useful) and understandable to the user. The rule "a customer with a loan has outstanding debt" is valid but trivial, so it is discarded in step 5.

Khai phá dữ liệu khác truy vấn SQL (môn Cơ sở dữ liệu) ở loại câu hỏi. Truy vấn trả lời câu hỏi người dùng đã biết cách diễn đạt; khai phá tìm điều người dùng chưa nghĩ tới để hỏi. Cùng một bảng khách hàng, câu truy vấn sau đếm số khách và tổng dư nợ theo hạng thẻ:

Data mining differs from an SQL query (the Databases course) in the kind of question. A query answers a question the user can already phrase; mining finds what the user has not thought to ask. On the same customer table, the following query counts customers and total debt by card tier:

```
SELECT hang_the, COUNT(*) AS so_khach, SUM(du_no) AS tong_no
FROM khach_hang
GROUP BY hang_the;
```

Tiêu chí <i>Criterion</i>	Truy vấn SQL <i>SQL query</i>	Thống kê mô tả <i>Descriptive statistics</i>	Khai phá dữ liệu <i>Data mining</i>
Câu hỏi <i>Question</i>	Đã biết, viết thành điều kiện <i>Known, written as a condition</i>	Đã biết, hỏi về đặc trưng của tập dữ liệu <i>Known, about the features of the data set</i>	Chưa biết trước; thuật toán đề xuất mẫu <i>Not known beforehand; the algorithm proposes patterns</i>
Kết quả <i>Result</i>	Các dòng hay số tổng hợp <i>Rows or aggregate figures</i>	Trung bình, độ phân tán, biểu đồ <i>Means, spread, charts</i>	Quy tắc, mô hình, nhóm, điểm dự đoán <i>Rules, models, groups, predicted scores</i>
Ví dụ ở Phú Sa <i>Example at Phú Sa</i>	Đếm khách hạng Vàng <i>Count Gold-tier customers</i>	Thu nhập trung vị 15 triệu đồng <i>Median income 15 million dong</i>	Dự đoán hồ sơ mới có thành nợ xấu không <i>Predict whether a new application becomes bad debt</i>

Mã sau chạy lại cả năm bước trên bảng 12 khách ở thu nhỏ: chọn ba cột, làm sạch bằng cách điền thu nhập thiếu bằng trung vị, biến đổi thu nhập thành ba mức, rồi khai phá bằng cách đếm khách có nhà theo mức.

The following code runs the five steps again on the 12-customer table in miniature: select three columns, clean by filling the missing income with the median, transform income into three levels, then mine by counting house owners per level.

```
import statistics

# (mã, thu nhập triệu/tháng, có nhà); KH07 thiếu thu nhập
du_lieu = [("KH01", 9, 0), ("KH02", 12, 0), ("KH03", 15, 1),
           ("KH04", 22, 1), ("KH05", 8, 0), ("KH06", 18, 1),
           ("KH07", None, 1), ("KH08", 14, 0), ("KH09", 11, 0),
           ("KH10", 16, 1), ("KH11", 20, 1), ("KH12", 120, 1)]
# Làm sạch: điền ô thiếu bằng trung vị của các ô có giá trị
```

```
co_gia_tri = [t for _, t, _ in du_lieu if t is not None]
tv = statistics.median(co_gia_tri)
sach = [(m, tv if t is None else t, h) for m, t, h in du_lieu]

# Biến đổi: rời rạc hóa thu nhập thành ba mức
def muc(t):
    return "Thấp" if t < 12 else "Trung bình" if t < 20 else "Cao"

# Khai phá: đếm khách có nhà theo mức thu nhập
dem = {}
for m, t, h in sach:
    n, c = dem.get(muc(t), (0, 0))
    dem[muc(t)] = (n + 1, c + h)
print("Trung vị dùng để điền:", tv)
for k in ("Thấp", "Trung bình", "Cao"):
    n, c = dem[k]
    print(f"{k:10s} {n} khách, {c} có nhà ({c / n:.0%})")
```

Kết quả chạy / Output

```
Trung vị dùng để điền: 15
Thấp      3 khách, 0 có nhà (0%)
Trung bình 6 khách, 4 có nhà (67%)
Cao       3 khách, 3 có nhà (100%)
```

Trung vị của 11 thu nhập có giá trị là 15 triệu đồng, nên ô thiếu của KH07 được điền 15. Kết quả cho một mẫu có thể đọc ra: tỷ lệ có nhà tăng theo mức thu nhập, 0 trên 3, 4 trên 6 (67%) và 3 trên 3. Với 12 khách, đây là gợi ý cần kiểm, chưa phải kết luận. Chương 2 và 3 tính từng bước này kỹ hơn.

The median of the 11 observed incomes is 15 million dong, so the missing cell of KH07 is filled with 15. The run yields a readable pattern: the house-ownership rate rises with the income level, 0 of 3, 4 of 6 (67%) and 3 of 3. With 12 customers this is a lead to be tested, not a conclusion. Chapters 2 and 3 work through each of these steps in more detail.

PHẢI THUỘC / MEMORISE — KDD là cả quá trình năm bước; khai phá dữ liệu là bước thứ tư. Mẫu có giá trị khi hợp lệ, mới, hữu ích và hiểu được.

KDD is the whole five-step process; data mining is the fourth step. A pattern has value when it is valid, novel, useful and understandable.

LỖI HAY GẶP / COMMON MISTAKE — Viết "khai phá dữ liệu gồm làm sạch, chọn, biến đổi và tìm mẫu". Đó là mô tả KDD. Khai phá dữ liệu theo nghĩa hẹp chỉ là bước chạy thuật toán tìm mẫu; các bước còn lại là chuẩn bị và đánh giá.

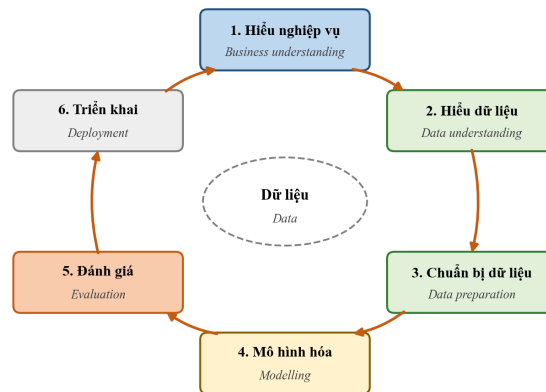
Writing "data mining consists of cleaning, selecting, transforming and finding patterns". That describes KDD. Data mining in the narrow sense is only the step that runs the pattern-finding algorithm; the other steps are preparation and evaluation.

1.3. CRISP-DM và dự án chấm điểm tín dụng của Phú Sa — 1.3. CRISP-DM and Phú Sa's credit scoring project

CRISP-DM (Cross-Industry Standard Process for Data Mining) là quy trình chuẩn cho dự án khai phá dữ liệu, gồm sáu pha xếp thành vòng. Khác với KDD tả luồng biến đổi của dữ liệu, CRISP-DM tả việc của nhóm dự án và nhấn mạnh nghiệp vụ ở đầu và cuối vòng.

Vòng lặp là chủ ý: gặp dữ liệu không đủ thì quay lại pha hiểu nghiệp vụ để đổi mục tiêu, mô hình chưa đạt thì quay lại pha chuẩn bị dữ liệu.

CRISP-DM (Cross-Industry Standard Process for Data Mining) is the standard process for a data mining project, with six phases arranged in a cycle. Where KDD describes how the data is transformed, CRISP-DM describes the work of the project team and puts the business at both the start and the end of the cycle. The loop is deliberate: when the data turns out to be insufficient the team returns to business understanding to change the goal, and when a model falls short it returns to data preparation.



Hình 1.3. Vòng CRISP-DM: sáu pha quanh dữ liệu, có thể quay lại pha trước

Figure 1.3. The CRISP-DM cycle: six phases around the data, with the option to return to an earlier phase

Áp vào Phú Sa: khối dữ liệu do giám đốc Phạm Thu Hà phụ trách xây mô hình chấm điểm tín dụng cho khoản vay tiền mặt trên ứng dụng "Phú Sa". Chuyên viên phân tích Đỗ Quang Huy dựng bảng dữ liệu và chạy mô hình; thực tập sinh Lê Ngọc Trâm kiểm đối chiếu nhãn nợ xấu với sổ thu hồi nợ. Bảng sau ghi việc của từng pha.

Applied to Phú Sa: the data division led by director Phạm Thu Hà builds a credit scoring model for cash loans on the "Phú Sa" mobile app. Analyst Đỗ Quang Huy builds the data table and runs the models; intern Lê Ngọc Trâm checks the bad-debt labels against the collections ledger. The table records the work of each phase.

Pha Phase	Việc chính Main tasks	Ở Phú Sa At Phú Sa	Chương g Chapter
1. Hiểu nghiệp vụ <i>1. Business understanding</i>	Nêu mục tiêu kinh doanh, tiêu chí thành công, ràng buộc <i>State the business goal, success criteria and constraints</i>	Giảm số khoản vay thành nợ xấu mà không từ chối nhiều khách tốt <i>Reduce loans turning into bad debt without rejecting many good customers</i>	1, 9

Pha <i>Phase</i>	Việc chính <i>Main tasks</i>	Ở Phú Sa <i>At Phú Sa</i>	Chương <i>Chapter</i>
2. Hiểu dữ liệu <i>2. Data understanding</i>	Thu thập, mô tả, soát chất lượng dữ liệu <i>Collect, describe and check the quality of the data</i>	Thống kê thu nhập, dư nợ; phát hiện thu nhập thiếu và ngoại lai <i>Summarise income and debt; find missing incomes and outliers</i>	2
3. Chuẩn bị dữ liệu <i>3. Data preparation</i>	Làm sạch, tích hợp, biến đổi, chọn thuộc tính, chia tập huấn luyện và kiểm tra <i>Clean, integrate, transform, select attributes, split into training and test sets</i>	Điền thu nhập thiếu, tính DTI, chuẩn hóa, chia tập hồ sơ <i>Fill missing income, compute DTI, normalise, split the applications</i>	3
4. Mô hình hóa <i>4. Modelling</i>	Chọn kỹ thuật, dựng và điều chỉnh mô hình <i>Choose techniques, build and tune models</i>	Cây quyết định, Naive Bayes, kNN, hồi quy logistic <i>Decision tree, Naive Bayes, kNN, logistic regression</i>	5, 6
5. Đánh giá <i>5. Evaluation</i>	Đo mô hình trên tập kiểm tra, đối chiếu với tiêu chí nghiệp vụ <i>Measure the model on the test set and compare with the business criteria</i>	Ma trận nhầm lẫn, ROC, chi phí bỏ sót nợ xấu <i>Confusion matrix, ROC, cost of missed bad debt</i>	7
6. Triển khai <i>6. Deployment</i>	Đưa mô hình vào hệ thống, giám sát, huấn luyện lại, tuân thủ pháp luật <i>Put the model into the system, monitor it, retrain it, comply with the law</i>	Chấm điểm hồ sơ trên ứng dụng, theo dõi độ ổn định điểm theo tháng <i>Score applications in the app and track score stability month by month</i>	7, 9

Chia dữ liệu thuộc pha 3 và phải làm trước khi dựng mô hình: mô hình học trên tập huấn luyện và chỉ được chấm điểm trên tập kiểm tra mà nó chưa thấy. Mã sau chia 14 hồ sơ vay thành tập kiểm tra 4 hồ sơ bằng bộ sinh số ngẫu nhiên có hạt giống 2026, nên lần chạy nào cũng ra cùng kết quả.

Splitting the data belongs to phase 3 and must be done before the model is built: the model learns on the training set and is scored only on a test set it has not seen. The following code splits the 14 loan applications into a test set of 4 using a random generator with seed 2026, so every run gives the same result.

```
import random

ma = [f"HS{i:02d}" for i in range(1, 15)]
nhan = dict(zip(ma, [1] * 5 + [0] * 9)) # 1 = nợ xấu
rng = random.Random(2026)
kiem_tra = sorted(rng.sample(ma, 4))
huan_luyen = [m for m in ma if m not in kiem_tra]
print("Kiểm tra:", kiem_tra)
print("Huấn luyện:", len(huan_luyen), "hồ sơ")
for ten, tap in (("huấn luyện", huan_luyen), ("kiểm tra", kiem_tra)):
    xau = sum(nhan[m] for m in tap)
    print(f"Nợ xấu trong tập {ten}: {xau}/{len(tap)}")
```

Kết quả chạy / Output

```
Kiểm tra: ['HS02', 'HS06', 'HS09', 'HS12']
Huấn luyện: 10 hồ sơ
Nợ xấu trong tập huấn luyện: 4/10
Nợ xấu trong tập kiểm tra: 1/4
```

Tập kiểm tra gồm HS02, HS06, HS09 và HS12; tập huấn luyện có 10 hồ sơ. Số hồ sơ nợ xấu là 4 trong tập huấn luyện và 1 trong tập kiểm tra, cộng lại đủ 5. Với tập nhỏ như vậy, phép chia ngẫu nhiên có thể làm lệch tỷ lệ lớp giữa hai tập; Chương 3 và 7 trình bày cách chia phân tầng.

The test set holds HS02, HS06, HS09 and HS12; the training set has 10 applications. Bad-debt applications number 4 in the training set and 1 in the test set, adding up to 5. With a set this small a random split can skew the class ratio between the two sets; Chapters 3 and 7 show stratified splitting.

PHẢI THUỘC / MEMORISE — Sáu pha CRISP-DM theo thứ tự: hiểu nghiệp vụ, hiểu dữ liệu, chuẩn bị dữ liệu, mô hình hóa, đánh giá, triển khai. Quá trình lặp, không đi một chiều. *The six CRISP-DM phases in order: business understanding, data understanding, data preparation, modelling, evaluation, deployment. The process iterates; it does not run in one direction.*

LỖI HAY GẶP / COMMON MISTAKE — Vào thẳng mô hình hóa mà chưa có tiêu chí thành công. Khi chưa nói "tốt" nghĩa là gì (giảm bao nhiêu nợ xấu, bỏ sót tối đa bao nhiêu), không thể quyết định mô hình nào đạt ở pha đánh giá. *Jumping into modelling without a success criterion. Until "good" has been defined (how much bad debt is to be cut, how many misses are tolerable), the evaluation phase cannot decide which model passes.*

1.4. Nhiệm vụ mô tả, nhiệm vụ dự đoán và các bài toán khai phá — 1.4.

Descriptive and predictive tasks and the mining problems

Nhiệm vụ khai phá chia làm hai nhóm. Nhiệm vụ mô tả (descriptive) tìm đặc trưng chung của dữ liệu hiện có: các nhóm khách giống nhau, các sản phẩm hay mua cùng nhau, điểm khác thường. Nhiệm vụ dự đoán (predictive) dùng dữ liệu đã biết kết quả để suy ra giá trị chưa biết của dữ liệu mới: khách này có trả nợ đúng hạn không, dư nợ tháng sau là bao nhiêu. Bảng sau xếp bảy bài toán thường gặp theo câu hỏi kinh doanh; thuật toán trong cột bốn được trình bày ở các chương ghi cột cuối.

Mining tasks fall into two groups. Descriptive tasks find general features of the existing data: groups of similar customers, products bought together, unusual points. Predictive tasks use data

with known outcomes to infer the unknown value for new data: whether this customer will repay on time, what next month's outstanding debt will be. The table sets out seven common problems by business question; the algorithms in column four are covered in the chapters named in the last column.

Câu hỏi kinh doanh <i>Business question</i>	Bài toán <i>Problem</i>	Thuật toán tiêu biểu <i>Typical algorithms</i>	Ví dụ ở Phú Sa <i>Example at Phú Sa</i>	Chương <i>Chapter</i>
Hồ sơ này có thành nợ xấu không? <i>Will this application become bad debt?</i>	Phân lớp <i>Classification</i>	Cây quyết định, Naive Bayes, kNN, hồi quy logistic <i>Decision tree, Naive Bayes, kNN, logistic regression</i>	Đoán nhãn Tốt hoặc Xấu cho hồ sơ vay mới <i>Predict the label Good or Bad for a new loan application</i>	5, 6
Dư nợ tháng sau là bao nhiêu? <i>What will next month's debt be?</i>	Hồi quy <i>Regression</i>	Hồi quy tuyến tính <i>Linear regression</i>	Dự báo dư nợ toàn công ty cho tháng 10/2026 <i>Forecast the company's outstanding debt for October 2026</i>	6
Khách nào giống khách nào? <i>Which customers resemble each other?</i>	Phân cụm <i>Clustering</i>	k-means, phân cấp, DBSCAN <i>k-means, hierarchical, DBSCAN</i>	Chia khách theo số giao dịch và chi tiêu <i>Group customers by transactions and spending</i>	8
Sản phẩm nào hay mua cùng nhau? <i>Which products are bought together?</i>	Luật kết hợp <i>Association rules</i>	Apriori, FP-growth <i>Apriori, FP-growth</i>	Khách trả góp xe máy đều mua kèm bảo hiểm khoản vay (3 trên 3 giao dịch) <i>Every motorbike instalment customer also buys loan insurance (3 of 3 transactions)</i>	4
Giao dịch nào khác thường? <i>Which transaction is unusual?</i>	Phát hiện bất thường <i>Anomaly detection</i>	z-score, IQR, khoảng cách, isolation forest <i>z-score, IQR, distance, isolation forest</i>	Giao dịch thẻ 4.200 nghìn đồng giữa các giao dịch vài trăm nghìn <i>A card transaction of 4,200 thousand dong among transactions of a few hundred thousand</i>	9

Câu hỏi kinh doanh <i>Business question</i>	Bài toán <i>Problem</i>	Thuật toán tiêu biểu <i>Typical algorithms</i>	Ví dụ ở Phú Sa <i>Example at Phú Sa</i>	Chương <i>Chapter</i>
Khách thường dùng sản phẩm theo thứ tự nào? <i>In what order do customers take products?</i>	Khai phá chuỗi <i>Sequence mining</i>	Mẫu tuần tự <i>Sequential patterns</i>	Vay tiền mặt, rồi thẻ, rồi bảo hiểm <i>Cash loan, then card, then insurance</i>	4
Khách nói gì về dịch vụ? <i>What do customers say about the service?</i>	Khai phá văn bản <i>Text mining</i>	TF-IDF, cosine, phân tích cảm xúc <i>TF-IDF, cosine, sentiment analysis</i>	Nhóm phản hồi về lãi suất và thủ tục <i>Group feedback about interest rates and procedures</i>	9

Hai bài toán đầu là dự đoán, năm bài toán sau là mô tả. Phân lớp và hồi quy giống nhau ở chỗ cần dữ liệu đã có đáp án; chúng khác nhau ở kiểu đáp án: phân lớp trả về một nhãn rời rạc, hồi quy trả về một số liên tục. Phân lớp và phân cụm thường bị nhầm vì kết quả của cả hai đều là các nhóm.

The first two problems are predictive and the other five are descriptive. Classification and regression are alike in needing data with known answers; they differ in the kind of answer: classification returns a discrete label, regression a continuous number. Classification and clustering are often confused because both produce groups.

Tiêu chí <i>Criterion</i>	Phân lớp <i>Classification</i>	Phân cụm <i>Clustering</i>
Nhãn có sẵn <i>Labels given</i>	Có, mỗi dòng huấn luyện có nhãn Tốt hoặc Xấu <i>Yes, each training row has the label Good or Bad</i>	Không, nhóm do thuật toán tạo ra <i>No, the groups are created by the algorithm</i>
Học theo kiểu <i>Learning type</i>	Có giám sát <i>Supervised</i>	Không giám sát <i>Unsupervised</i>
Đầu ra <i>Output</i>	Nhãn cho dòng mới; có thể đo đúng hay sai <i>A label for a new row; right or wrong can be measured</i>	Phân hoạch dữ liệu thành cụm; chất lượng đo gián tiếp <i>A partition of the data into clusters; quality measured indirectly</i>
Ví dụ ở Phú Sa <i>Example at Phú Sa</i>	Hồ sơ vay mới là Tốt hay Xấu <i>Is a new loan application Good or Bad</i>	Chia khách theo mức giao dịch và chi tiêu rồi đặt tên nhóm <i>Group customers by activity and spending, then name the groups</i>

PHẢI THUỘC / MEMORISE — Phân lớp và hồi quy cần nhãn (có giám sát), một trả nhãn rời rạc, một trả số. Phân cụm, luật kết hợp, phát hiện bất thường không cần nhãn.
Classification and regression need labels (supervised), one returning a discrete label and the other a number. Clustering, association rules and anomaly detection need no labels.

LỖI HAY GẶP / COMMON MISTAKE — Dùng phân lớp khi chưa có nhãn. Nếu Phú Sa chưa ghi nhận khoản vay nào là nợ xấu hay không, không có dữ liệu huấn luyện để phân lớp; việc đầu tiên là định nghĩa và gắn nhãn, hoặc dùng phân cụm để khám phá.
Using classification when no labels exist. If Phú Sa has not recorded which loans became bad debt, there is no training data for classification; the first job is to define and attach the labels, or to use clustering to explore.

1.5. Học có giám sát, không giám sát và bán giám sát — 1.5. Supervised, unsupervised and semi-supervised learning

Cách thuật toán học phụ thuộc vào việc dữ liệu có nhãn hay không. Học có giám sát (supervised learning) dùng các dòng đã biết đáp án, gọi là nhãn, để học quy tắc từ thuộc tính ra nhãn; phân lớp và hồi quy thuộc nhóm này. Học không giám sát (unsupervised learning) chỉ có thuộc tính, không có nhãn, và tìm cấu trúc trong dữ liệu: phân cụm, luật kết hợp, phát hiện bất thường. Học bán giám sát (semi-supervised learning) dùng một phần nhỏ dòng có nhãn cùng nhiều dòng chưa có nhãn.

How an algorithm learns depends on whether the data is labelled. Supervised learning uses rows with a known answer, called the label, to learn a rule from attributes to label; classification and regression belong here. Unsupervised learning has attributes only, no labels, and looks for structure in the data: clustering, association rules, anomaly detection. Semi-supervised learning uses a small share of labelled rows together with many unlabelled ones.

Phú Sa gặp cả ba tình huống. Hồ sơ vay đã đáo hạn có nhãn Tốt hoặc Xấu, đủ để học có giám sát. Dữ liệu giao dịch thẻ không có nhãn gian lận sẵn nên phải dùng phương pháp không giám sát để tìm giao dịch khác thường. Khoản vay vừa giải ngân chưa biết sẽ thành nợ xấu hay không, nên là dòng chưa có nhãn; khi nhãn khan hiếm và việc gắn nhãn tốn công, học bán giám sát tận dụng cả hai loại dòng.

Phú Sa meets all three situations. Loan applications that have matured carry the label Good or Bad, enough for supervised learning. Card transaction data has no ready-made fraud label, so unsupervised methods are needed to find unusual transactions. A loan just disbursed is not yet known to become bad debt, so it is an unlabelled row; when labels are scarce and costly to attach, semi-supervised learning makes use of both kinds of rows.

Tiêu chí Criterion	Có giám sát Supervised	Không giám sát Unsupervised	Bán giám sát Semi-supervised
Dữ liệu Data	Mọi dòng có nhãn Every row labelled	Không có nhãn No labels	Ít dòng có nhãn, nhiều dòng không Few rows labelled, many not

Tiêu chí <i>Criterion</i>	Có giám sát <i>Supervised</i>	Không giám sát <i>Unsupervised</i>	Bán giám sát <i>Semi-supervised</i>
Mục tiêu <i>Goal</i>	Đoán nhãn cho dòng mới <i>Predict the label of a new row</i>	Tìm nhóm, quy luật, điểm lạ <i>Find groups, rules, odd points</i>	Đoán nhãn, dùng cấu trúc của dòng chưa nhãn <i>Predict labels, using the structure of unlabelled rows</i>
Thuật toán <i>Algorithms</i>	Cây quyết định, kNN, hồi quy <i>Decision tree, kNN, regression</i>	k-means, Apriori, DBSCAN <i>k-means, Apriori, DBSCAN</i>	Tự gán nhãn, lan truyền nhãn <i>Self-training, label propagation</i>
Ví dụ ở Phú Sa <i>Example at Phú Sa</i>	Đoán nợ xấu từ hồ sơ đã đáo hạn <i>Predict bad debt from matured applications</i>	Chia khách theo hành vi dùng ứng dụng <i>Group customers by app behaviour</i>	Khoản vay chưa đủ kỳ hạn để biết nhãn <i>Loans too young for the label to be known</i>

Ví dụ nhỏ nhất của học có giám sát là quy tắc một ngưỡng. Mười hồ sơ của bộ dữ liệu ở Chương 6 có thu nhập, tỷ lệ nợ trên thu nhập (DTI, tính bằng phần trăm) và nhãn Tốt hay Xấu. Quy tắc: DTI lớn hơn 40 thì đoán Xấu, ngược lại đoán Tốt. Mã sau áp quy tắc cho cả mười hồ sơ, đếm bốn ô của bảng đúng sai và so với cách luôn đoán lớp đông nhất.

The smallest example of supervised learning is a one-threshold rule. The ten applications of the Chapter 6 data set carry income, debt-to-income ratio (DTI, in percent) and the label Good or Bad. The rule: DTI above 40 means predict Bad, otherwise predict Good. The following code applies the rule to all ten, counts the four cells of the right-and-wrong table and compares with always guessing the largest class.

Bài mẫu 2. Chấm quy tắc ngưỡng DTI trên 40 — *Worked example 2. Scoring the rule DTI above 40*

```
# (mã, thu nhập triệu/tháng, DTI %, nhãn thật)
td = [("TD01", 9, 52, "Xấu"), ("TD02", 11, 46, "Xấu"),
      ("TD03", 14, 40, "Tốt"), ("TD04", 16, 30, "Tốt"),
      ("TD05", 20, 25, "Tốt"), ("TD06", 26, 18, "Tốt"),
      ("TD07", 13, 44, "Xấu"), ("TD08", 18, 38, "Xấu"),
      ("TD09", 22, 33, "Tốt"), ("TD10", 10, 35, "Tốt")]

def quy_tac(dti):
    return "Xấu" if dti > 40 else "Tốt"

bang = {"TP": 0, "FN": 0, "FP": 0, "TN": 0}
dung = 0
for ma, thu, dti, that in td:
    doan = quy_tac(dti)
    dung += doan == that
    if that == "Xấu":
        bang["TP" if doan == "Xấu" else "FN"] += 1
    else:
        bang["FP" if doan == "Xấu" else "TN"] += 1
    if doan != that:
        print("Sai:", ma, "DTI", dti, "đoán", doan, "thật", that)
print(bang)
```

```
print(f"Đúng {dung}/{len(td)} = {dung / len(td):.0%}")
nhan = [x[3] for x in td]
print("Luôn đoán Tốt:", nhan.count("Tốt") / len(td))
```

Kết quả chạy / Output

```
Sai: TD08 DTI 38 đoán Tốt thật Xấu
{'TP': 3, 'FN': 1, 'FP': 0, 'TN': 6}
Đúng 9/10 = 90%
Luôn đoán Tốt: 0.6
```

Mã Code	DTI (%)	Nhân thật Actual	Quy tắc đoán Rule predicts	Kết quả Outcome
TD01	52	Xấu <i>Bad</i>	Xấu <i>Bad</i>	Đúng <i>Right</i>
TD02	46	Xấu <i>Bad</i>	Xấu <i>Bad</i>	Đúng <i>Right</i>
TD03	40	Tốt <i>Good</i>	Tốt <i>Good</i>	Đúng <i>Right</i>
TD04	30	Tốt <i>Good</i>	Tốt <i>Good</i>	Đúng <i>Right</i>
TD05	25	Tốt <i>Good</i>	Tốt <i>Good</i>	Đúng <i>Right</i>
TD06	18	Tốt <i>Good</i>	Tốt <i>Good</i>	Đúng <i>Right</i>
TD07	44	Xấu <i>Bad</i>	Xấu <i>Bad</i>	Đúng <i>Right</i>
TD08	38	Xấu <i>Bad</i>	Tốt <i>Good</i>	Sai <i>Wrong</i>
TD09	33	Tốt <i>Good</i>	Tốt <i>Good</i>	Đúng <i>Right</i>
TD10	35	Tốt <i>Good</i>	Tốt <i>Good</i>	Đúng <i>Right</i>

- Với từng hồ sơ so DTI với 40: TD01 (52), TD02 (46) và TD07 (44) lớn hơn 40 nên bị đoán Xấu; bảy hồ sơ còn lại bị đoán Tốt (TD03 có DTI đúng bằng 40, không lớn hơn).

For each application compare DTI with 40: TD01 (52), TD02 (46) and TD07 (44) exceed 40 and are predicted Bad; the other seven are predicted Good (TD03 has a DTI of exactly 40, not above it).

- So với nhân thật: ba hồ sơ bị đoán Xấu đều thật sự Xấu; sáu hồ sơ Tốt đều được đoán Tốt; riêng TD08 (DTI 38) thật sự Xấu nhưng bị đoán Tốt.

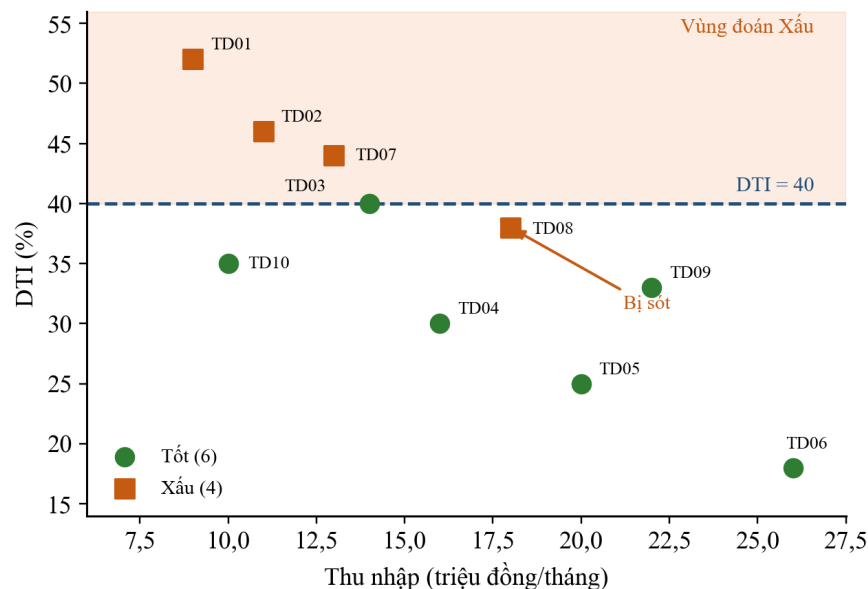
Compared with the actual labels: the three predicted Bad are all truly Bad; all six Good applications are predicted Good; only TD08 (DTI 38) is truly Bad but predicted Good.

3. Gọi Xấu là lớp cần tìm: TP = 3 (đoán Xấu, thật Xấu), FN = 1 (đoán Tốt, thật Xấu), FP = 0 (đoán Xấu, thật Tốt), TN = 6.

Take Bad as the class to find: TP = 3 (predicted Bad, truly Bad), FN = 1 (predicted Good, truly Bad), FP = 0 (predicted Bad, truly Good), TN = 6.

4. Độ chính xác = $(TP + TN) / 10 = (3 + 6) / 10 = 90\%$. Cách luôn đoán Tốt đúng 6/10 = 60% (đúng 0,6 trong kết quả chạy). Quy tắc hơn 30 điểm phần trăm.

Accuracy = $(TP + TN) / 10 = (3 + 6) / 10 = 90\%$. Always guessing Good is right 6/10 = 60% (0.6 in the run output). The rule is ahead by 30 percentage points.



Hình 1.4. Mười hồ sơ theo thu nhập và DTI; vùng tô là nơi quy tắc đoán Xấu, TD08 là hồ sơ Xấu bị sót

Figure 1.4. Ten applications by income and DTI; the shaded area is where the rule predicts Bad, and TD08 is the Bad application the rule misses

Độ chính xác 90% đo trên chính mười hồ sơ dùng để chọn ngưỡng, nên lạc quan. Ngưỡng 36 cũng cho 90% trên mười hồ sơ này (bỏ sót 0 hồ sơ Xấu nhưng đoán nhầm TD03), tức dữ liệu chưa đủ để phân biệt hai ngưỡng. Chương 6 và 7 trình bày cách chọn ngưỡng và đo trên tập kiểm tra tách riêng.

The 90% accuracy is measured on the same ten applications used to choose the threshold, so it is optimistic. A threshold of 36 also gives 90% on these ten (missing no Bad application but misjudging TD03), so the data cannot separate the two thresholds. Chapters 6 and 7 show how to choose a threshold and measure on a separate test set.

PHẢI THUỘC / MEMORISE — Có giám sát: có nhãn, học ánh xạ từ thuộc tính ra nhãn. Không giám sát: không nhãn, tìm cấu trúc. Bán giám sát: ít nhãn, nhiều dòng chưa nhãn.
Supervised: labels exist, learn the mapping from attributes to label. Unsupervised: no labels, find structure. Semi-supervised: few labels, many unlabelled rows.

LỖI HAY GẶP / COMMON MISTAKE — Báo độ chính xác đo trên chính dữ liệu đã dùng để chọn ngưỡng hay dựng mô hình. Con số 90% ở trên chỉ nói quy tắc khớp với mười

hồ sơ; khả năng đoán hồ sơ mới phải đo trên tập kiểm tra.

Reporting accuracy measured on the same data used to choose the threshold or build the model. The 90% above says only that the rule fits the ten applications; the ability to predict new applications must be measured on a test set.

1.6. Các loại dữ liệu được khai phá — 1.6. Types of data that are mined

Dữ liệu có cấu trúc (structured) nằm trong bảng có cột và kiểu cố định; dữ liệu bán cấu trúc (semi-structured) như JSON, XML có thể nhưng độ dài và thứ tự thay đổi; dữ liệu phi cấu trúc (unstructured) như văn bản tự do, ảnh, âm thanh không có khung sẵn. Theo cách tổ chức, năm loại sau gặp nhiều trong khai phá dữ liệu tài chính.

Structured data sits in tables with fixed columns and types; semi-structured data such as JSON and XML has tags but variable length and order; unstructured data such as free text, images and audio has no ready frame. By how it is organised, the following five types are common in financial data mining.

Loại Type	Cách tổ chức Organisation	Ví dụ ở Phú Sa Example at Phú Sa	Bài toán hay dùng Typical problem	Chương Chapter
Bảng quan hệ <i>Relational table</i>	Mỗi dòng là một đối tượng, mỗi cột là một thuộc tính <i>Each row an object, each column an attribute</i>	Bảng 12 khách hàng; 14 hồ sơ vay <i>The 12-customer table; the 14 loan applications</i>	Phân lớp, phân cụm, hồi quy <i>Classification, clustering, regression</i>	2 đến 8
Giao dịch <i>Transaction</i>	Mỗi bản ghi là một tập mục đi cùng nhau <i>Each record is a set of items occurring together</i>	10 giao dịch, mỗi giao dịch là các sản phẩm khách dùng <i>10 transactions, each the products a customer uses</i>	Luật kết hợp <i>Association rules</i>	4
Chuỗi thời gian <i>Time series</i>	Giá trị đo lặp theo thứ tự thời gian <i>Values measured repeatedly in time order</i>	Dư nợ cuối tháng từ 10/2025 đến 9/2026 <i>Month-end debt from October 2025 to September 2026</i>	Dự báo, phát hiện bất thường <i>Forecasting, anomaly detection</i>	6, 9
Văn bản <i>Text</i>	Chuỗi từ, cần tách từ và vector hóa <i>Sequences of words, requiring tokenisation and vectorisation</i>	Bốn phản hồi của khách về lãi suất và thủ tục <i>Four customer comments on interest rates and procedures</i>	Phân nhóm, tìm tương tự, cảm xúc <i>Grouping, similarity, sentiment</i>	9

Loại Type	Cách tổ chức Organisation	Ví dụ ở Phú Sa Example at Phú Sa	Bài toán hay dùng Typical problem	Chương Chapter
Đồ thị Graph	Đỉnh là thực thể, cạnh là quan hệ <i>Nodes are entities, edges are relationships</i>	Hai khách cùng đăng nhập từ một thiết bị là hai đỉnh nối nhau <i>Two customers logging in from one device are two linked nodes</i>	Phát hiện gian lận, lan truyền rủi ro <i>Fraud detection, risk propagation</i>	9

Giao dịch và chuỗi thời gian có mã chạy ngay ở đây. Dữ liệu giao dịch ghi mỗi lần khách mua gói sản phẩm; mã sau đếm tần suất từng sản phẩm trong 10 giao dịch (BH: bảo hiểm khoản vay; TD: thẻ tín dụng; VM: vay tiền mặt; DM: trả góp điện máy; XM: trả góp xe máy).

Transactions and time series have code that runs here. Transaction data records each time a customer takes a bundle of products; the following code counts how often each product appears in 10 transactions (BH: loan insurance; TD: credit card; VM: cash loan; DM: electronics instalment; XM: motorbike instalment).

```
from collections import Counter

giao_dich = [
    {"TD", "VM", "BH"}, {"TD", "BH"}, {"VM", "DM", "BH"},
    {"TD", "VM", "BH"}, {"XM", "BH"}, {"TD", "DM"},
    {"VM", "XM", "BH"}, {"TD", "VM"}, {"DM", "XM", "BH"},
    {"TD", "VM", "BH"},
]
n = len(giao_dich)
dem = Counter(sp for g in giao_dich for sp in g)
for sp, c in sorted(dem.items(), key=lambda kv: (-kv[1], kv[0])):
    print(f"{sp}: {c}/{n} = {c / n:.0%}")
tb = sum(len(g) for g in giao_dich) / n
print("Số sản phẩm mỗi giao dịch:", tb)
```

Kết quả chạy / Output

```
BH: 8/10 = 80%
TD: 6/10 = 60%
VM: 6/10 = 60%
DM: 3/10 = 30%
XM: 3/10 = 30%
Số sản phẩm mỗi giao dịch: 2.6
```

Bảo hiểm khoản vay có mặt trong 8 trong 10 giao dịch (80%), thẻ và vay tiền mặt mỗi loại 60%, hai loại trả góp mỗi loại 30%. Mỗi giao dịch có trung bình 2,6 sản phẩm. Dữ liệu này không có cột thuộc tính, chỉ có tập mục, nên cần thuật toán riêng (Chương 4).

Loan insurance appears in 8 of 10 transactions (80%), the card and the cash loan 60% each, the two instalment products 30% each. Each transaction holds 2.6 products on average. This data has no attribute columns, only item sets, so it needs its own algorithms (Chapter 4).

Chuỗi thời gian có thứ tự là thông tin: đổi chỗ hai tháng là làm mất xu hướng. Mã sau đọc 12 giá trị dư nợ cuối tháng (tỷ đồng) và tính mức thay đổi từng tháng.

A time series carries information in its order: swapping two months destroys the trend. The following code reads 12 month-end debt values (billion dong) and computes the change in each month.

```
thang = ["10/2025", "11/2025", "12/2025", "01/2026", "02/2026",
        "03/2026", "04/2026", "05/2026", "06/2026", "07/2026",
        "08/2026", "09/2026"]
du_no = [820.5, 838.2, 851.7, 879.3, 884.1, 912.6,
        925.8, 948.2, 951.7, 987.4, 1003.9, 1030.5] # tỷ đồng
chenh = [round(b - a, 1) for a, b in zip(du_no, du_no[1:])]
print("Chênh lệch từng tháng (tỷ đồng):")
print(" ", chenh[:6])
print(" ", chenh[6:])
print("Tăng cả kỳ:", round(du_no[-1] - du_no[0], 1))
print(f"Tăng tương đối: {du_no[-1] / du_no[0] - 1:.1%}")
print("Trung bình mỗi tháng:", round(sum(chenh) / len(chenh), 2))
lon = max(chenh)
print("Tháng tăng nhiều nhất:", thang[chenh.index(lon) + 1], lon)
```

Kết quả chạy / Output

```
Chênh lệch từng tháng (tỷ đồng):
[17.7, 13.5, 27.6, 4.8, 28.5, 13.2]
[22.4, 3.5, 35.7, 16.5, 26.6]
Tăng cả kỳ: 210.0
Tăng tương đối: 25.6%
Trung bình mỗi tháng: 19.09
Tháng tăng nhiều nhất: 07/2026 35.7
```

Dư nợ tăng từ 820,5 lên 1.030,5 tỷ đồng, thêm 210,0 tỷ đồng (25,6%) trong 11 bước tháng, trung bình 19,09 tỷ đồng mỗi tháng. Tháng tăng nhiều nhất là 07/2026 với 35,7 tỷ đồng. Chuỗi luôn đi lên nhưng không đều, nên dự báo cần mô hình xu hướng (Chương 6) và phép làm trơn (Chương 9).

Debt rises from 820.5 to 1,030.5 billion dong, an increase of 210.0 billion dong (25.6%) over 11 monthly steps, 19.09 billion dong a month on average. The largest monthly rise is July 2026 at 35.7 billion dong. The series always rises but unevenly, so forecasting needs a trend model (Chapter 6) and smoothing (Chapter 9).

Dữ liệu lớn (big data) là dữ liệu mà khối lượng, tốc độ hoặc độ đa dạng vượt khả năng của công cụ thông thường. Mô hình 3V gồm khối lượng (volume), tốc độ (velocity), đa dạng (variety); mô hình 5V thêm độ tin cậy (veracity) và giá trị (value).

Big data is data whose volume, speed or variety exceeds what ordinary tools can handle. The 3V model covers volume, velocity and variety; the 5V model adds veracity and value.

V	Ý nghĩa Meaning	Biểu hiện ở Phú Sa How it shows at Phú Sa
Khối lượng Volume	Số bản ghi và dung lượng lớn <i>A large number of records and a large size</i>	Mỗi lần mở ứng dụng, xem khoản vay, bấm thanh toán đều thành một bản ghi sự kiện <i>Every app opening, loan view and payment tap becomes an event record</i>
Tốc độ Velocity	Dữ liệu đến liên tục, cần xử lý nhanh <i>Data arrives continuously and must be processed fast</i>	Chấm điểm hồ sơ ngay khi khách nộp trên ứng dụng <i>Scoring an application the moment the customer submits it in the app</i>

V	Ý nghĩa <i>Meaning</i>	Biểu hiện ở Phú Sa <i>How it shows at Phú Sa</i>
Đa dạng <i>Variety</i>	Nhiều dạng: bảng, văn bản, ảnh, chuỗi sự kiện <i>Many forms: tables, text, images, event streams</i>	Bảng khoản vay, phản hồi tự do, ảnh giấy tờ định danh <i>Loan tables, free-text feedback, images of identity papers</i>
Tin cậy <i>Veracity</i>	Mức đúng, đủ, nhất quán của dữ liệu <i>How accurate, complete and consistent the data is</i>	Thu nhập tự khai thiếu hoặc sai, mã khách trùng <i>Self-declared income missing or wrong, duplicated customer codes</i>
Giá trị <i>Value</i>	Dữ liệu phải đem lại lợi ích kinh doanh đo được <i>The data must bring a measurable business benefit</i>	Tiền nợ xấu tránh được nhờ chấm điểm tốt hơn <i>Bad debt avoided by better scoring</i>

PHẢI THUỘC / MEMORISE — Ba V gốc: khối lượng, tốc độ, đa dạng; hai V thêm: độ tin cậy, giá trị. Dữ liệu giao dịch là tập mục, chuỗi thời gian có thứ tự là thông tin, văn bản cần tách từ.

The three original Vs: volume, velocity, variety; the two added Vs: veracity, value. Transaction data is a set of items, a time series carries information in its order, text needs tokenising.

LỖI HAY GẶP / COMMON MISTAKE — Xáo trộn hay chia ngẫu nhiên chuỗi thời gian như dữ liệu bảng. Tháng tương lai lọt vào tập huấn luyện làm kết quả đẹp giả; với chuỗi thời gian, tập kiểm tra phải là các tháng sau tập huấn luyện.

Shuffling or randomly splitting a time series as if it were a table. Future months leaking into the training set produce a falsely good result; for a time series the test set must be the months after the training set.

1.7. Kho dữ liệu, OLAP và sự khác biệt với khai phá dữ liệu — 1.7. Data warehouse, OLAP and how they differ from data mining

Hệ thống giao dịch (OLTP, online transaction processing) ghi từng nghiệp vụ như giải ngân, thu nợ; dữ liệu thay đổi liên tục và tối ưu cho ghi nhanh. Kho dữ liệu (data warehouse) là nơi gom dữ liệu lịch sử từ nhiều hệ thống OLTP để phân tích, có bốn đặc điểm: hướng chủ đề (subject-oriented), tích hợp (integrated), theo thời gian (time-variant) và ổn định (non-volatile, ghi thêm chứ không sửa). Việc đưa dữ liệu vào kho gọi là ETL (extract, transform, load: trích, biến đổi, nạp), tương ứng bước làm sạch và tích hợp của KDD.

A transaction system (OLTP, online transaction processing) records each operation such as a disbursement or a repayment; its data changes continuously and is tuned for fast writes. A data warehouse gathers historical data from several OLTP systems for analysis and has four features: subject-oriented, integrated, time-variant and non-volatile (appended to, not edited). Loading data into the warehouse is called ETL (extract, transform, load), matching the cleaning and integration step of KDD.

OLAP (online analytical processing) là cách hỏi đáp nhanh trên kho dữ liệu theo các chiều. Chiều (dimension) là góc nhìn như nghề, hạng thẻ, tháng; số đo (measure) là con số được tổng hợp như số khách, dư nợ. Năm phép thao tác thường gặp: roll-up (gộp lên mức tổng quát hơn, ví dụ từ tháng lên quý), drill-down (xuống mức chi tiết hơn), slice (cắt một giá trị của một chiều), dice (cắt theo nhiều chiều), pivot (xoay trục để đổi cách nhìn).

OLAP (online analytical processing) is fast question answering on a warehouse along dimensions. A dimension is a viewpoint such as occupation, card tier or month; a measure is a summarised figure such as customer count or outstanding debt. Five common operations: roll-up (aggregate to a more general level, for example from month to quarter), drill-down (go to a finer level), slice (fix one value of one dimension), dice (select on several dimensions), pivot (rotate the axes to change the view).

Mã sau dựng khối hai chiều nghề và hạng thẻ từ 12 khách, in số khách từng ô rồi thực hiện roll-up theo nghề để có dư nợ theo hạng thẻ (triệu đồng).

The following code builds a two-dimensional cube of occupation and card tier from the 12 customers, prints the customer count in each cell, then rolls up over occupation to get debt by card tier (million dong).

```
from collections import Counter

khach = [ # (nghề, hạng thẻ, dư nợ triệu đồng)
    ("Công nhân", "Chuẩn", 12), ("Nhân viên văn phòng", "Chuẩn", 18),
    ("Giáo viên", "Vàng", 20), ("Kinh doanh tự do", "Vàng", 45),
    ("Công nhân", "Chuẩn", 10), ("Nhân viên văn phòng", "Vàng", 30),
    ("Kinh doanh tự do", "Vàng", 40), ("Giáo viên", "Chuẩn", 16),
    ("Nhân viên văn phòng", "Chuẩn", 14), ("Công nhân", "Vàng", 25),
    ("Giáo viên", "Bạch kim", 35), ("Kinh doanh tự do", "Bạch kim", 150),
]
hang = ["Chuẩn", "Vàng", "Bạch kim"]
nghe = sorted({k[0] for k in khach})
dem = Counter({k[0], k[1]} for k in khach)
print(f'Nghề:22s' + " ".join(f'{h:>10s}' for h in hang))
for ng in nghe:
    print(f'{ng:22s}' + " ".join(f'{dem[(ng, h)]:>10d}' for h in hang))
# Roll-up: gộp mọi nghề, cộng dư nợ theo hạng thẻ
no = Counter()
for _, h, d in khach:
    no[h] += d
print("Dư nợ theo hạng:", {h: no[h] for h in hang})
print("Tổng dư nợ:", sum(no.values()), "triệu đồng")
```

Kết quả chạy / Output

Nghề	Chuẩn	Vàng	Bạch kim
Công nhân	2	1	0
Giáo viên	1	1	1
Kinh doanh tự do	0	2	1
Nhân viên văn phòng	2	1	0
Dư nợ theo hạng:	{'Chuẩn': 70, 'Vàng': 160, 'Bạch kim': 185}		
Tổng dư nợ:	415 triệu đồng		

Hạng Bạch kim có 2 trong 12 khách (16,7%) nhưng giữ 185 trong 415 triệu đồng dư nợ (44,6%). Đó là câu trả lời OLAP: nó cho biết dư nợ tập trung ở đâu. Nó không nói khách Bạch kim nào sẽ trả chậm; câu hỏi đó cần mô hình dự đoán ở Chương 5 và 6.

The Platinum tier has 2 of 12 customers (16.7%) but holds 185 of 415 million dong of debt (44.6%). That is an OLAP answer: it says where the debt is concentrated. It does not say which Platinum customer will pay late; that question needs the predictive models of Chapters 5 and 6.

Tiêu chí <i>Criterion</i>	OLAP <i>OLAP</i>	Khai phá dữ liệu <i>Data mining</i>
Hướng tiếp cận <i>Approach</i>	Người dùng chọn chiều và phép toán, hệ thống tổng hợp <i>The user chooses dimensions and operations; the system aggregates</i>	Thuật toán tự tìm mẫu từ dữ liệu <i>The algorithm finds patterns from the data itself</i>
Câu hỏi <i>Question</i>	Bao nhiêu, ở đâu, khi nào <i>How many, where, when</i>	Vì sao, ai sẽ, nhóm nào, cái gì đi cùng cái gì <i>Why, who will, which group, what goes with what</i>
Kết quả <i>Result</i>	Bảng tổng hợp, khối dữ liệu <i>Summary tables, cubes</i>	Quy tắc, mô hình, cụm, điểm dự đoán <i>Rules, models, clusters, predicted scores</i>
Ví dụ ở Phú Sa <i>Example at Phú Sa</i>	Dư nợ theo hạng thẻ và quý <i>Debt by card tier and quarter</i>	Khách nào có xác suất nợ xấu cao <i>Which customers have a high probability of bad debt</i>

PHẢI THUỘC / MEMORISE — OLAP tổng hợp theo chiều do người dùng chọn và trả lời "bao nhiêu, ở đâu". Khai phá dữ liệu tự tìm mẫu và trả lời "vì sao, ai sẽ". Kho dữ liệu là nguồn thuận lợi cho cả hai nhưng không bắt buộc phải có.

OLAP aggregates along dimensions the user chooses and answers "how many, where". Data mining finds patterns itself and answers "why, who will". A warehouse is a convenient source for both but is not mandatory.

LỖI HAY GẶP / COMMON MISTAKE — Cho rằng có kho dữ liệu và biểu đồ OLAP là đã khai phá dữ liệu. Biểu đồ roll-up chỉ tổng hợp; thiếu bước tìm mẫu và kiểm mẫu trên dữ liệu mới thì chưa phải khai phá.

Believing that having a warehouse and OLAP charts amounts to data mining. A roll-up chart only aggregates; without a step that finds patterns and tests them on new data it is not mining.

1.8. Ứng dụng trong tài chính và công cụ khai phá dữ liệu — 1.8. Applications in finance and data mining tools

Sáu ứng dụng sau là các bài toán Phú Sa gặp hằng ngày; mỗi bài toán quy về một hay nhiều bài toán ở mục 1.4.

The following six applications are problems Phú Sa meets every day; each reduces to one or more of the problems in section 1.4.

Ứng dụng <i>Application</i>	Câu hỏi <i>Question</i>	Bài toán <i>Problem</i>	Dữ liệu dùng <i>Data used</i>	Chương <i>Chapter</i>
Chấm điểm tín dụng <i>Credit scoring</i>	Khách này có trả nợ đúng hạn không? <i>Will this customer repay on time?</i>	Phân lớp, xác suất <i>Classification, probability</i>	Thu nhập, DTI, lịch sử trả nợ <i>Income, DTI, repayment history</i>	5, 6, 7
Chống gian lận <i>Fraud prevention</i>	Giao dịch hay hồ sơ nào khác thường? <i>Which transaction or application is unusual?</i>	Phát hiện bất thường <i>Anomaly detection</i>	Giá trị, thời điểm, thiết bị giao dịch <i>Value, time and device of transactions</i>	9
Bán chéo <i>Cross-selling</i>	Khách này nên được mời sản phẩm nào? <i>Which product should this customer be offered?</i>	Luật kết hợp <i>Association rules</i>	Giỏ sản phẩm của từng khách <i>Each customer's product basket</i>	4
Phân khúc khách hàng <i>Customer segmentation</i>	Có mấy nhóm khách, mỗi nhóm cần chính sách nào? <i>How many customer groups are there, and what policy does each need?</i>	Phân cụm, RFM <i>Clustering, RFM</i>	Số giao dịch, chi tiêu, thời gian từ lần cuối <i>Number of transactions, spending, time since the last one</i>	8
Dự báo dòng tiền <i>Cash-flow forecasting</i>	Dư nợ và tiền thu tháng tới ra sao? <i>What will next month's debt and collections be?</i>	Hồi quy, chuỗi thời gian <i>Regression, time series</i>	Dư nợ, tiền thu theo tháng <i>Monthly debt and collections</i>	6, 9
Phòng chống rửa tiền <i>Anti-money laundering</i>	Giao dịch nào cần chuyên viên tuân thủ kiểm tra? <i>Which transaction needs a compliance officer's review?</i>	Bất thường, luật cảnh báo <i>Anomalies, alert rules</i>	Chuỗi giao dịch, quan hệ giữa các tài khoản <i>Transaction sequences, links between accounts</i>	9

Ứng dụng cuối có người trong vòng quyết định: mô hình chỉ đề xuất cảnh báo, chuyên viên tuân thủ quyết định có báo cáo hay không. Ngưỡng cảnh báo đặt thấp thì bắt nhiều giao dịch lạ nhưng tạo nhiều cảnh báo sai, đặt cao thì ngược lại; đánh đổi này nói tiếp ở Chương 7 và 9.

The last application keeps a person in the loop: the model only proposes alerts and a compliance officer decides whether to report. A low alert threshold catches many unusual transactions but raises many false alerts, a high one does the reverse; this trade-off is taken up in Chapters 7 and 9.

Về công cụ, sách này viết thuật toán thuần Python bằng thư viện chuẩn để người học tính tay và kiểm lại từng bước. Trong dự án thật, nhóm dùng thư viện có sẵn; mỗi công cụ ở bảng sau chỉ được mô tả cách dùng, không so sánh hơn kém.

As for tools, this book writes the algorithms in plain Python using the standard library so that learners can compute by hand and check each step. In a real project the team uses ready-made libraries; each tool in the table is described by how it is used, not ranked.

Công cụ <i>Tool</i>	Loại <i>Kind</i>	Cách dùng <i>How it is used</i>	Thành phần nhắc đến trong sách <i>Components named in this book</i>
Python, scikit-learn, pandas	Ngôn ngữ lập trình và thư viện <i>Programming language and libraries</i>	Viết mã dựng quy trình từ làm sạch đến mô hình; pandas giữ bảng dữ liệu, scikit-learn cung cấp thuật toán <i>Code builds the pipeline from cleaning to model; pandas holds the data table, scikit-learn supplies algorithms</i>	DecisionTreeClassifier, GaussianNB, KMeans
R	Ngôn ngữ thống kê <i>Statistical language</i>	Phân tích thống kê và dựng mô hình bằng gói và hàm viết sẵn <i>Statistical analysis and modelling with ready-made packages and functions</i>	Các gói thống kê và phân cụm <i>Statistics and clustering packages</i>
Weka	Phần mềm có giao diện đồ họa, viết bằng Java <i>Graphical software written in Java</i>	Nạp tệp dữ liệu, chọn thuật toán trong menu, đọc kết quả mà không cần viết mã <i>Load a data file, pick an algorithm from a menu, read the results without writing code</i>	Cây quyết định, Naive Bayes, kNN, k-means, Apriori <i>Decision tree, Naive Bayes, kNN, k-means, Apriori</i>
Orange	Phần mềm lập trình trực quan <i>Visual programming software</i>	Kéo và nối các khối thao tác thành quy trình, có thể mở rộng bằng Python <i>Drag and connect operation blocks into a workflow, extensible with Python</i>	Phân lớp, phân cụm, biểu đồ khám phá <i>Classification, clustering, exploratory charts</i>

PHẢI THUỘC / MEMORISE — Mỗi ứng dụng tài chính quy về bài toán khai phá: chấm điểm là phân lớp, chống gian lận và rửa tiền là phát hiện bất thường, bán chéo là luật kết hợp, phân khúc là phân cụm, dòng tiền là hồi quy và chuỗi thời gian.

Each financial application reduces to a mining problem: scoring is classification, fraud and money-laundering control is anomaly detection, cross-selling is association rules, segmentation is clustering, cash flow is regression and time series.

LỖI HAY GẶP / COMMON MISTAKE — Chọn công cụ trước rồi mới tìm bài toán. Công cụ chỉ thực thi thuật toán; bài toán và tiêu chí đánh giá phải xác định từ pha hiểu nghiệp vụ

(mục 1.3).

Choosing the tool first and the problem afterwards. A tool only executes algorithms; the problem and the evaluation criterion must come from the business understanding phase (section 1.3).

1.9. Quy trình dự án, vai trò nhân sự và thách thức — 1.9. Project process, team roles and challenges

Một dự án khai phá dữ liệu ở công ty tài chính chạy theo vòng CRISP-DM nhưng được quản lý như mọi dự án khác: có phạm vi, mốc bàn giao, người chịu trách nhiệm từng pha. Lặp lại thường quay về pha 3 khi dữ liệu chưa đủ sạch, và vòng đầu thường chỉ cần một mô hình đơn giản làm mốc so sánh. Sáu vai trò sau tham gia dự án chấm điểm tín dụng của Phú Sa.

A data mining project at a finance company runs on the CRISP-DM cycle but is managed like any other project: it has a scope, delivery milestones and an owner for each phase. Iterations often return to phase 3 when the data is not yet clean enough, and the first cycle usually needs only a simple model as a baseline. The following six roles take part in Phú Sa's credit scoring project.

Vai trò <i>Role</i>	Việc chính <i>Main work</i>	Ở Phú Sa <i>At Phú Sa</i>
Chủ nghiệp vụ <i>Business owner</i>	Đặt mục tiêu, tiêu chí thành công, nhận kết quả vào quy trình duyệt vay <i>Sets the goal and success criteria, takes the result into the lending process</i>	Lãnh đạo khối tín dụng <i>Head of the credit division</i>
Lãnh đạo khối dữ liệu <i>Head of data</i>	Phân việc, giữ tiến độ, báo cáo chủ nghiệp vụ <i>Assigns work, keeps the schedule, reports to the business owner</i>	Giám đốc Phạm Thu Hà <i>Director Phạm Thu Hà</i>
Kỹ sư dữ liệu <i>Data engineer</i>	Xây luồng ETL, kho dữ liệu, bảo đảm dữ liệu cập nhật <i>Builds the ETL flow and warehouse, keeps data up to date</i>	Nhóm hạ tầng dữ liệu <i>The data infrastructure team</i>
Nhà phân tích dữ liệu <i>Data analyst</i>	Khám phá dữ liệu, chuẩn bị bảng, dựng và đánh giá mô hình <i>Explores the data, prepares the table, builds and evaluates models</i>	Chuyên viên Đỗ Quang Huy; thực tập sinh Lê Ngọc Trâm hỗ trợ kiểm nhãn <i>Analyst Đỗ Quang Huy; intern Lê Ngọc Trâm helps check labels</i>
Chuyên viên tuân thủ <i>Compliance officer</i>	Soát cơ sở pháp lý xử lý dữ liệu cá nhân, nội dung xin đồng ý <i>Reviews the legal basis for processing personal data and the consent wording</i>	Bộ phận pháp chế, tuân thủ <i>The legal and compliance unit</i>

Vai trò <i>Role</i>	Việc chính <i>Main work</i>	Ở Phú Sa <i>At Phú Sa</i>
Kiểm định mô hình <i>Model validator</i>	Kiểm độc lập mô hình trước khi đưa vào dùng, theo dõi sau triển khai <i>Independently checks the model before use and monitors it after deployment</i>	Bộ phận quản trị rủi ro <i>The risk management unit</i>

Năm thách thức sau lặp lại ở hầu hết dự án. Mỗi thách thức có biểu hiện cụ thể ở Phú Sa và một cách xử lý.

The following five challenges recur in most projects. Each has a concrete form at Phú Sa and a way of handling it.

Thách thức <i>Challenge</i>	Biểu hiện ở Phú Sa <i>How it shows at Phú Sa</i>	Cách xử lý <i>How to handle it</i>	Chương <i>Chapter</i>
Chất lượng dữ liệu <i>Data quality</i>	Thu nhập của KH07 thiếu; thu nhập 120 triệu đồng của KH12 cách xa nhóm còn lại <i>KH07's income is missing; KH12's income of 120 million dong lies far from the rest</i>	Soát, điền hoặc loại; kiểm ngoại lai theo quy tắc rõ ràng <i>Check, fill or drop; test outliers by an explicit rule</i>	2, 3
Quyền riêng tư <i>Privacy</i>	Hồ sơ vay chứa thông tin tài chính, tín dụng của khách <i>Loan files hold customers' financial and credit information</i>	Chỉ lấy cột cần dùng, dùng mã khách thay tên, giới hạn người truy cập <i>Take only the columns needed, use customer codes instead of names, limit access</i>	9
Giải thích được <i>Explainability</i>	Khách bị từ chối hỏi lý do; nhân viên cần nói được mô hình dựa vào đâu <i>A rejected customer asks why; staff must say what the model relied on</i>	Ưu tiên mô hình đọc được như cây quyết định, hồi quy; ghi lại các yếu tố ảnh hưởng <i>Prefer readable models such as trees and regression; record the factors that drive the score</i>	5, 6, 9
Quá khớp <i>Overfitting</i>	Mô hình đúng 99% trên hồ sơ cũ nhưng sai nhiều trên hồ sơ mới <i>A model 99% right on old applications but often wrong on new ones</i>	Chia tập huấn luyện và kiểm tra, kiểm chéo <i>Split into training and test sets, cross-validate</i>	7
Thiên lệch <i>Bias</i>	Nhãn nợ xấu chỉ có ở khách đã được duyệt, khách bị từ chối không có nhãn <i>Bad-debt labels exist only for approved customers; rejected ones have none</i>	Nhận diện nhóm thiếu dữ liệu, kiểm tác động theo nhóm khách <i>Identify the groups with missing data and test the effect by customer group</i>	9

PHẢI THUỘC / MEMORISE — Năm thách thức: chất lượng dữ liệu, quyền riêng tư, giải thích được, quá khớp, thiên lệch. Dự án chạy theo vòng, chủ nghiệp vụ đặt tiêu chí và người kiểm định độc lập chấm mô hình.

Five challenges: data quality, privacy, explainability, overfitting, bias. The project runs in a cycle; the business owner sets the criteria and an independent validator scores the model.

LỖI HAY GẶP / COMMON MISTAKE — Chỉ giao dự án cho người phân tích, bỏ chủ nghiệp vụ và pháp chế khỏi cuộc họp đầu. Mô hình đẹp về kỹ thuật nhưng không trả lời đúng câu hỏi kinh doanh, hoặc dùng cột dữ liệu mà công ty không được phép dùng, phải làm lại từ đầu.

Giving the project to the analyst alone and leaving the business owner and legal staff out of the first meeting. A model that is technically good but does not answer the business question, or uses a data column the company may not use, has to be redone from the start.

1.10. Khung pháp lý tóm lược — 1.10. A short legal framework

Khai phá dữ liệu khách hàng là xử lý dữ liệu, và với công ty tài chính còn là xử lý dữ liệu cá nhân và thông tin khách hàng. Bảng sau tóm tắt bốn nhóm quy định nền tảng, mỗi dòng ghi Điều đã đối chiếu với văn bản. Chi tiết, kèm Luật Trí tuệ nhân tạo, để Chương 9.

Mining customer data is data processing, and for a finance company also the processing of personal data and customer information. The table summarises four groups of basic rules, each row naming the Article checked against the text. Details, together with the Law on Artificial Intelligence, are left to Chapter 9.

Văn bản <i>Instrument</i>	Điều <i>Article</i>	Nội dung <i>Content</i>	Liên hệ với dự án <i>Link to the project</i>
Luật Dữ liệu 60/2024/QH15, hiệu lực 01/7/2025 (bản hợp nhất 42/VBHN-VPQH ngày 16/3/2026) <i>Law on Data 60/2024/QH15, effective 01/7/2025 (consolidated text 42/VBHN-VPQH of 16/3/2026)</i>	Điều 3 khoản 1, 8, 9 <i>Article 3, clauses 1, 8, 9</i>	Dữ liệu là dữ liệu số về sự vật, hiện tượng, sự kiện; xử lý dữ liệu gồm tiếp nhận, chuyển đổi, tổ chức dữ liệu và các hoạt động khác; cơ sở dữ liệu là tập hợp dữ liệu được sắp xếp để truy cập, khai thác <i>Data means digital data on things, phenomena and events; data processing covers receiving, converting and organising data and other activities; a database is a set of data arranged for access and exploitation</i>	Bảng khách hàng, kho dữ liệu và các bước làm sạch, biến đổi đều là xử lý dữ liệu <i>The customer table, the warehouse and the cleaning and transformation steps are all data processing</i>
Luật Bảo vệ dữ liệu cá nhân 91/2025/QH15,	Điều 2 khoản 1, 3, 11; Điều 3	Dữ liệu cá nhân gồm dữ liệu cơ bản và nhạy cảm, sau khi khử nhận dạng không còn là dữ liệu cá nhân;	Chỉ dùng dữ liệu khách cho mục đích đã nêu; mã khách còn bảng đối

Văn bản <i>Instrument</i>	Điều <i>Article</i>	Nội dung <i>Content</i>	Liên hệ với dự án <i>Link to the project</i>
hiệu lực 01/01/2026 <i>Law on Personal Data Protection 91/2025/QH15, effective 01/01/2026</i>	khoản 2 <i>Article 2, clauses 1, 3, 11; Article 3, clause 2</i>	dữ liệu nhạy cảm thuộc danh mục do Chính phủ ban hành; chỉ thu thập, xử lý đúng phạm vi, mục đích cụ thể, rõ ràng <i>Personal data comprises basic and sensitive data and stops being personal data once de-identified; sensitive data is in a list issued by the Government; collection and processing must stay within a specific and clear scope and purpose</i>	chỉ cho người thật chưa phải là khử nhận dạng <i>Use customer data only for the stated purpose; a customer code that still has a lookup table to a real person is not de-identification</i>
Nghị định 356/2025/NĐ-CP (31/12/2025, hiệu lực 01/01/2026; Nghị định 13/2023/NĐ-CP hết hiệu lực) <i>Decree 356/2025/ND-CP (31/12/2025, effective 01/01/2026; Decree 13/2023/ND-CP ceased to be in force)</i>	Điều 4 khoản 1 điểm k; Điều 8 khoản 2 điểm a; Điều 42 khoản 2 <i>Article 4, clause 1(k); Article 8, clause 2(a); Article 42, clause 2</i>	Thông tin tài chính, tín dụng và lịch sử giao dịch của khách tại tổ chức tín dụng là dữ liệu cá nhân nhạy cảm; khi xin đồng ý, tổ chức tài chính, ngân hàng phải nêu rõ mục đích, gồm chấm điểm, xếp hạng tín dụng nếu có <i>Financial and credit information and transaction history of customers at credit institutions is sensitive personal data; when seeking consent, financial and banking organisations must state the purposes, including credit scoring and rating where applicable</i>	Dữ liệu khoản vay của Phú Sa là dữ liệu nhạy cảm; nội dung xin đồng ý phải nói rõ có chấm điểm tín dụng <i>Phú Sa's loan data is sensitive data; the consent wording must say that credit scoring takes place</i>
Luật Các tổ chức tín dụng, bản hợp nhất 143/2026/VBH N-LQ-VPQH ngày 23/9/2026 <i>Law on Credit Institutions, consolidated text 143/2026/VBHN-LQ-VPQH of 23/9/2026</i>	Điều 4 khoản 41; Điều 13 khoản 1, 2, 3 <i>Article 4, clause 41; Article 13, clauses 1, 2, 3</i>	Công ty tài chính là loại hình tổ chức tín dụng phi ngân hàng. Người quản lý, nhân viên không được tiết lộ thông tin khách hàng; tổ chức tín dụng phải bảo đảm bí mật thông tin khách hàng và không cung cấp cho tổ chức, cá nhân khác, trừ yêu cầu của cơ quan nhà nước có thẩm quyền theo luật hoặc được khách hàng chấp thuận <i>A finance company is a type of non-bank credit institution. Managers and staff may not disclose customer information; a credit institution must keep customer information confidential and may not provide it to other organisations or individuals,</i>	Gửi dữ liệu khách cho nhà thầu hay công cụ bên ngoài công ty cần có căn cứ này; nhân viên không được tiết lộ thông tin khách <i>Sending customer data to a contractor or a tool outside the company needs this basis; staff may not disclose customer information</i>

Văn bản <i>Instrument</i>	Điều <i>Article</i>	Nội dung <i>Content</i>	Liên hệ với dự án <i>Link to the project</i>
		<i>except at the request of a competent state body under law or with the customer's consent</i>	

Hai điểm cần nhớ khi làm bài. Thứ nhất, Nghị định 356/2025 thay Nghị định 13/2023, nên không viện dẫn văn bản cũ như đang hiệu lực. Thứ hai, bản hợp nhất 143/2026 đã gộp Luật 23/2026/QH16 nhưng phần do Luật này sửa chỉ có hiệu lực từ 01/12/2026; Điều 13 nêu ở bảng là nội dung của bản hợp nhất hiện hành.

Two points to remember in an exam. First, Decree 356/2025 replaces Decree 13/2023, so the old text is not cited as in force. Second, the consolidated text 143/2026 has merged Law 23/2026/QH16 but the parts amended by that Law take effect only on 01/12/2026; Article 13 in the table is the content of the current consolidated text.

PHẢI THUỘC / MEMORISE — Thông tin tài chính, tín dụng của khách tại tổ chức tín dụng là dữ liệu cá nhân nhạy cảm (Nghị định 356/2025 Điều 4 khoản 1 điểm k). Tổ chức tín dụng phải giữ bí mật thông tin khách hàng (Luật Các tổ chức tín dụng Điều 13).
Financial and credit information of customers at credit institutions is sensitive personal data (Decree 356/2025, Article 4, clause 1(k)). A credit institution must keep customer information confidential (Law on Credit Institutions, Article 13).

LỖI HAY GẶP / COMMON MISTAKE — Gọi mọi bảng có mã khách là dữ liệu đã ẩn danh. Theo Luật Bảo vệ dữ liệu cá nhân Điều 2 khoản 11, khử nhận dạng là làm cho dữ liệu không thể xác định hoặc không thể giúp xác định một con người cụ thể; mã khách vẫn nói được về người thật qua bảng đối chiếu thì chưa đạt.
Calling every table with customer codes anonymised data. Under the Law on Personal Data Protection, Article 2, clause 11, de-identification makes data unable to identify or help identify a specific person; a customer code that can still be linked back to a real person through a lookup table does not qualify.

1.11. Bảng thuật ngữ — 1.11. Glossary

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Dữ liệu	data	Giá trị ghi lại về đối tượng hay sự kiện, chưa gắn câu hỏi <i>A recorded value about an object or event, not yet tied to a question</i>
Thông tin	information	Dữ liệu đã tổng hợp, đặt trong ngữ cảnh để trả lời một câu hỏi <i>Data summarised and placed in context to answer a question</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Tri thức	knowledge	Mẫu hay quy luật khái quát được cho dữ liệu mới <i>A pattern or rule that generalises to new data</i>
Khai phá dữ liệu	data mining	Dùng thuật toán tìm mẫu và mô hình có ích trong dữ liệu lớn <i>Using algorithms to find useful patterns and models in large data</i>
Khám phá tri thức trong cơ sở dữ liệu	knowledge discovery in databases (KDD)	Quá trình năm bước từ dữ liệu nguồn đến tri thức <i>The five-step process from source data to knowledge</i>
CRISP-DM	CRISP-DM	Quy trình chuẩn sáu pha cho dự án khai phá dữ liệu <i>The standard six-phase process for a data mining project</i>
Mô hình	model	Biểu diễn toán học học từ dữ liệu, dùng để dự đoán hoặc mô tả <i>A mathematical representation learned from data, used to predict or describe</i>
Thuộc tính	attribute	Một cột mô tả đối tượng, ví dụ thu nhập, nghề <i>A column describing an object, for example income or occupation</i>
Nhãn	label	Đáp án đã biết của một dòng, ví dụ Tốt hoặc Xấu <i>The known answer for a row, for example Good or Bad</i>
Phân lớp	classification	Dự đoán nhãn rời rạc cho dòng mới từ dòng đã có nhãn <i>Predicting a discrete label for a new row from labelled rows</i>
Hồi quy	regression	Dự đoán một giá trị số liên tục <i>Predicting a continuous numeric value</i>
Phân cụm	clustering	Chia dữ liệu thành nhóm giống nhau khi chưa có nhãn <i>Dividing data into similar groups when no labels exist</i>
Luật kết hợp	association rule	Quy tắc nói mục này hay xuất hiện cùng mục kia <i>A rule saying one item tends to occur with another</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Phát hiện bất thường	anomaly detection	Tìm điểm dữ liệu khác hẳn phần còn lại <i>Finding data points that differ sharply from the rest</i>
Học có giám sát	supervised learning	Học từ dòng có nhãn để đoán nhãn cho dòng mới <i>Learning from labelled rows to predict labels for new rows</i>
Học không giám sát	unsupervised learning	Tìm cấu trúc trong dữ liệu không có nhãn <i>Finding structure in data without labels</i>
Học bán giám sát	semi-supervised learning	Học từ ít dòng có nhãn cùng nhiều dòng chưa nhãn <i>Learning from few labelled rows plus many unlabelled ones</i>
Tập huấn luyện	training set	Phần dữ liệu để mô hình học <i>The part of the data the model learns from</i>
Tập kiểm tra	test set	Phần dữ liệu mô hình chưa thấy, dùng để chấm điểm <i>The part of the data the model has not seen, used for scoring</i>
Kho dữ liệu	data warehouse	Kho gom dữ liệu lịch sử từ nhiều hệ thống để phân tích <i>A store gathering historical data from several systems for analysis</i>
OLAP	online analytical processing (OLAP)	Tổng hợp nhanh theo nhiều chiều trên kho dữ liệu <i>Fast multidimensional aggregation on a warehouse</i>
ETL	extract, transform, load (ETL)	Trích, biến đổi, nạp dữ liệu vào kho <i>Extracting, transforming and loading data into the warehouse</i>
Dữ liệu lớn	big data	Dữ liệu vượt khả năng công cụ thường về khối lượng, tốc độ hay đa dạng <i>Data beyond ordinary tools in volume, velocity or variety</i>
Khử nhận dạng	de-identification	Thay đổi hoặc xóa thông tin để dữ liệu không còn xác định được con người cụ thể <i>Changing or removing information so data can no longer identify a specific person</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Dữ liệu cá nhân nhạy cảm	sensitive personal data	Dữ liệu cá nhân gắn với quyền riêng tư, thuộc danh mục do Chính phủ ban hành <i>Personal data tied to privacy, in a list issued by the Government</i>

1.12. Bài tập vận dụng có lời giải — *Applied exercises with solutions*

Bài 1. Quy tắc theo kỳ hạn vay — *Exercise 1. A rule based on loan term*

Công ty Tài chính TNHH Phú Sa ghi lại 8 khoản trả góp điện máy (dữ kiện đề bài): C01 đến C04 kỳ hạn 6 tháng, trong đó chỉ C03 trễ hạn; C05 đến C08 kỳ hạn 12 tháng, trong đó C05, C07 và C08 trễ hạn. (a) Tính tỷ lệ trễ hạn của từng kỳ hạn và của cả 8 khoản. (b) Lập quy tắc dùng một thuộc tính là kỳ hạn, mỗi kỳ hạn đoán theo lớp chiếm đa số. (c) Tính độ chính xác của quy tắc và so với cách luôn đoán lớp đông nhất.

Phú Sa Finance Company Limited records 8 electronics instalment loans (problem data): C01 to C04 have a 6-month term and only C03 is late; C05 to C08 have a 12-month term and C05, C07 and C08 are late. (a) Compute the late rate of each term and of all 8 loans. (b) Form a one-attribute rule on the term, predicting each term by its majority class. (c) Compute the rule's accuracy and compare it with always guessing the largest class.

- Kỳ hạn 6 tháng có 4 khoản, 1 khoản trễ: $1/4 = 25\%$. Kỳ hạn 12 tháng có 4 khoản, 3 khoản trễ: $3/4 = 75\%$. Cả 8 khoản có $1 + 3 = 4$ khoản trễ: $4/8 = 50\%$.
The 6-month term has 4 loans, 1 late: $1/4 = 25\%$. The 12-month term has 4 loans, 3 late: $3/4 = 75\%$. All 8 loans have $1 + 3 = 4$ late: $4/8 = 50\%$.
- Quy tắc: kỳ hạn 6 tháng đoán không trễ (lớp đa số, 3 trên 4); kỳ hạn 12 tháng đoán trễ (lớp đa số, 3 trên 4).
Rule: the 6-month term is predicted not late (majority class, 3 of 4); the 12-month term is predicted late (majority class, 3 of 4).
- Số khoản đoán đúng: $3 + 3 = 6$ trên 8, độ chính xác $6/8 = 75\%$.
Correct predictions: $3 + 3 = 6$ out of 8, an accuracy of $6/8 = 75\%$.
- Lớp đông nhất: 4 khoản trễ và 4 khoản không trễ bằng nhau, nên luôn đoán một lớp đúng $4/8 = 50\%$. Quy tắc theo kỳ hạn hơn 25 điểm phần trăm; với 8 khoản, kết luận này cần kiểm trên khoản vay mới.
The largest class: 4 late and 4 not late are tied, so always guessing one class is right $4/8 = 50\%$. The term-based rule is ahead by 25 percentage points; with 8 loans the conclusion needs testing on new loans.

Bài 2. Làm sạch và rời rạc hóa thu nhập — *Exercise 2. Cleaning and discretising income*

Bảng 7 khách có thu nhập (triệu đồng mỗi tháng, dữ kiện đề bài) là 10, 14, thiếu, 8, 20, 12, 16. (a) Điền ô thiếu bằng trung vị của các ô có giá trị. (b) Rời rạc hóa thành ba mức: dưới 12 là Thấp, từ 12 đến dưới 18 là Trung bình, từ 18 trở lên là Cao. (c) Đếm số khách mỗi mức và nói bước này thuộc bước nào của KDD.

A table of 7 customers has incomes (million dong per month, problem data) of 10, 14, missing, 8, 20, 12, 16. (a) Fill the missing cell with the median of the observed cells. (b) Discretise into three levels: under 12 is Low, from 12 up to under 18 is Medium, 18 and above is High. (c) Count the customers in each level and say which KDD step each operation belongs to.

1. Các ô có giá trị sắp tăng dần: 8, 10, 12, 14, 16, 20. Có 6 giá trị (số chẵn), trung vị là trung bình hai giá trị giữa: $(12 + 14) / 2 = 13$. Ô thiếu được điền 13.
The observed cells in ascending order: 8, 10, 12, 14, 16, 20. There are 6 values (an even count), so the median is the mean of the two middle values: $(12 + 14) / 2 = 13$. The missing cell is filled with 13.
2. Dãy đã điền: 10, 14, 13, 8, 20, 12, 16. Gán mức: 10 là Thấp; 14 là Trung bình; 13 là Trung bình; 8 là Thấp; 20 là Cao; 12 là Trung bình; 16 là Trung bình.
The filled series: 10, 14, 13, 8, 20, 12, 16. Assign levels: 10 is Low; 14 is Medium; 13 is Medium; 8 is Low; 20 is High; 12 is Medium; 16 is Medium.
3. Đếm: Thấp 2 khách, Trung bình 4 khách, Cao 1 khách, tổng 7. Mức Trung bình chiếm $4/7 = 57,1\%$.
Count: Low 2 customers, Medium 4, High 1, total 7. The Medium level holds $4/7 = 57.1\%$.
4. Điền giá trị thiếu thuộc bước 2 (làm sạch, tích hợp); rời rạc hóa thuộc bước 3 (biến đổi, thu giảm); đếm theo mức là việc của bước 4 (khai phá) ở dạng đơn giản nhất.
Filling the missing value belongs to step 2 (clean, integrate); discretisation to step 3 (transform, reduce); counting by level is step 4 (mining) in its simplest form.

Bài 3. Dung lượng dữ liệu sự kiện ứng dụng — Exercise 3. Storage size of app event data

Mỗi ngày có 40.000 khách mở ứng dụng "Phú Sa"; mỗi khách sinh 5 bản ghi sự kiện; mỗi bản ghi dài 200 byte (dữ kiện đề bài, 1 MB = 1.000.000 byte). (a) Tính số bản ghi và dung lượng mỗi ngày. (b) Tính dung lượng sau 30 ngày và sau 365 ngày. (c) Chỉ ra chữ V nào của dữ liệu lớn thể hiện rõ nhất và loại dữ liệu theo cách tổ chức.

Each day 40,000 customers open the "Phú Sa" app; each customer generates 5 event records; each record is 200 bytes long (problem data, 1 MB = 1,000,000 bytes). (a) Compute the number of records and the storage per day. (b) Compute the storage after 30 days and after 365 days. (c) Name the V of big data that shows most clearly and the type of data by organisation.

1. Số bản ghi mỗi ngày: $40.000 * 5 = 200.000$ bản ghi.
Records per day: $40,000 * 5 = 200,000$ records.
2. Dung lượng mỗi ngày: $200.000 * 200 = 40.000.000$ byte = 40 MB.
Storage per day: $200,000 * 200 = 40,000,000$ bytes = 40 MB.
3. Sau 30 ngày: $40 * 30 = 1.200$ MB = 1,2 GB. Sau 365 ngày: $40 * 365 = 14.600$ MB = 14,6 GB (1 GB = 1.000 MB).
After 30 days: $40 * 30 = 1,200$ MB = 1.2 GB. After 365 days: $40 * 365 = 14,600$ MB = 14.6 GB (1 GB = 1,000 MB).
4. Chữ V rõ nhất là tốc độ (velocity): bản ghi đến liên tục suốt ngày. Theo cách tổ chức, đây là chuỗi sự kiện, dạng dữ liệu bán cấu trúc. Dung lượng 14,6 GB mỗi năm vừa một máy tính; "lớn" ở đây nằm ở nhịp đến của dữ liệu hơn là kích thước.
The clearest V is velocity: records arrive continuously all day. By organisation this is an

event stream, semi-structured data. 14.6 GB a year fits on one computer; the "big" here lies in the pace of arrival more than in the size.

1.13. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines

Câu 1. Phân biệt dữ liệu, thông tin và tri thức; phân biệt KDD với khai phá dữ liệu. Minh họa bằng bảng khách hàng và bảng hồ sơ vay của Phú Sa.

Question 1. *Distinguish data, information and knowledge; distinguish KDD from data mining. Illustrate with Phú Sa's customer table and loan application table.*

- Dữ liệu là giá trị thô (dòng của KH03); thông tin là dữ liệu đã tổng hợp trả lời một câu hỏi (7 trong 12 khách có nhà); tri thức là mẫu khái quát dùng cho dữ liệu mới.
Data is a raw value (the row of KH03); information is summarised data answering a question (7 of 12 customers own a house); knowledge is a general pattern usable on new data.
- Ví dụ tri thức: trong 14 hồ sơ, lịch sử Xấu đi cùng tỷ lệ nợ xấu 80%, lịch sử Tốt đi cùng 0%; quy tắc theo lịch sử đúng 12/14 hồ sơ (85,7%).
A knowledge example: among 14 applications a Bad history goes with an 80% bad-debt rate and a Good history with 0%; the rule by history is right on 12 of 14 (85.7%).
- KDD là cả quá trình năm bước: chọn dữ liệu, làm sạch và tích hợp, biến đổi và thu giảm, khai phá, đánh giá và trình bày.
KDD is the whole five-step process: select data, clean and integrate, transform and reduce, mine, evaluate and present.
- Khai phá dữ liệu theo nghĩa hẹp chỉ là bước thứ tư, chạy thuật toán tìm mẫu; các bước còn lại chuẩn bị và đánh giá.
Data mining in the narrow sense is only the fourth step, running the pattern-finding algorithm; the other steps prepare and evaluate.
- Mẫu có giá trị khi hợp lệ, mới, hữu ích và hiểu được; tri thức phải kiểm trên dữ liệu chưa dùng để lập nó.
A pattern has value when valid, novel, useful and understandable; knowledge must be tested on data not used to form it.

Câu 2. Trình bày sáu pha CRISP-DM áp vào dự án chấm điểm tín dụng của Phú Sa và giải thích vì sao quy trình mang tính lặp.

Question 2. *Describe the six CRISP-DM phases applied to Phú Sa's credit scoring project and explain why the process is iterative.*

- Pha 1 hiểu nghiệp vụ: mục tiêu giảm khoản vay thành nợ xấu mà không từ chối nhiều khách tốt, kèm tiêu chí thành công.
Phase 1 business understanding: the goal of cutting loans that become bad debt without rejecting many good customers, with success criteria.
- Pha 2 hiểu dữ liệu: thống kê thu nhập, dư nợ, phát hiện giá trị thiếu (KH07) và ngoại lai (KH12).
Phase 2 data understanding: summarise income and debt, find the missing value (KH07) and the outlier (KH12).

- Pha 3 chuẩn bị dữ liệu: điền thiếu, tính DTI, chuẩn hóa, chia tập huấn luyện và kiểm tra.

Phase 3 data preparation: fill gaps, compute DTI, normalise, split into training and test sets.

- Pha 4 mô hình hóa, pha 5 đánh giá trên tập kiểm tra theo tiêu chí nghiệp vụ, pha 6 triển khai và giám sát độ ổn định điểm.

Phase 4 modelling, phase 5 evaluation on the test set against the business criteria, phase 6 deployment and monitoring of score stability.

- Quy trình lặp vì kết quả của một pha có thể làm đổi pha trước: dữ liệu không đủ thì đổi mục tiêu, mô hình chưa đạt thì quay lại chuẩn bị dữ liệu.

The process iterates because the result of one phase can change an earlier one: insufficient data changes the goal, a model that falls short sends the team back to data preparation.

Câu 3. So sánh OLAP với khai phá dữ liệu; nêu các thách thức và quy định pháp luật về thông tin khách hàng khi Phú Sa làm dự án chấm điểm tín dụng.

Question 3. Compare OLAP with data mining; state the challenges and the legal rules on customer information when Phú Sa runs a credit scoring project.

- OLAP tổng hợp theo chiều do người dùng chọn, trả lời "bao nhiêu, ở đâu" (dư nợ theo hạng thẻ); khai phá dữ liệu tự tìm mẫu, trả lời "vì sao, ai sẽ".

OLAP aggregates along dimensions the user chooses and answers "how many, where" (debt by card tier); data mining finds patterns itself and answers "why, who will".

- Thách thức: chất lượng dữ liệu, quyền riêng tư, giải thích được, quá khớp, thiên lệch; mỗi thách thức nêu biểu hiện ở Phú Sa và cách xử lý.

Challenges: data quality, privacy, explainability, overfitting, bias; for each give how it shows at Phú Sa and how to handle it.

- Luật Các tổ chức tín dụng (bản hợp nhất 143/2026) Điều 13: nhân viên không tiết lộ thông tin khách hàng; không cung cấp cho tổ chức, cá nhân khác, trừ yêu cầu của cơ quan có thẩm quyền theo luật hoặc khách hàng chấp thuận.

Law on Credit Institutions (consolidated text 143/2026) Article 13: staff may not disclose customer information; no provision to other organisations or individuals except at the request of a competent body under law or with the customer's consent.

- Luật Bảo vệ dữ liệu cá nhân 91/2025 Điều 2 khoản 1, 3, 11 và Nghị định 356/2025 Điều 4 khoản 1 điểm k: thông tin tài chính, tín dụng là dữ liệu nhạy cảm; khử nhận dạng làm dữ liệu không còn xác định được người.

Law on Personal Data Protection 91/2025 Article 2, clauses 1, 3, 11 and Decree 356/2025 Article 4, clause 1(k): financial and credit information is sensitive data; de-identification leaves data unable to identify a person.

- Nghị định 356/2025 Điều 8 khoản 2 điểm a: khi xin đồng ý phải nêu rõ mục đích, gồm chấm điểm, xếp hạng tín dụng nếu có.

Decree 356/2025 Article 8, clause 2(a): when seeking consent the purposes must be stated, including credit scoring and rating where applicable.

1.14. Câu hỏi trắc nghiệm — Multiple-choice questions

1. Trong bảng khách hàng của Phú Sa, phát biểu nào là tri thức theo nghĩa khai phá dữ liệu?

In Phú Sa's customer table, which statement is knowledge in the data mining sense?

- A. Dòng ghi KH03 là 34 tuổi, giáo viên, hạng thẻ Vàng, dư nợ 20 / *The row KH03 records age 34, teacher, Gold tier, debt 20*
- B. Bảng có 12 khách hàng, mỗi nghề có đúng 3 khách hàng / *The table has 12 customers, exactly 3 customers in each occupation*
- C. Tổng dư nợ của 12 khách hàng là 415 triệu đồng / *The total debt of the 12 customers is 415 million dong*
- D. Lịch sử Tốt đi cùng nợ xấu thấp, dùng để sàng hồ sơ mới / *A Good history goes with low bad debt, used to screen new applications*

2. Dựng bảng huấn luyện từ nhiều nguồn, điền giá trị thiếu và chia tập huấn luyện, tập kiểm tra thuộc pha nào của CRISP-DM?

Building the training table from several sources, filling missing values and splitting training and test sets belong to which CRISP-DM phase?

- A. Hiểu nghiệp vụ / *Business understanding*
- B. Hiểu dữ liệu / *Data understanding*
- C. Mô hình hóa / *Modelling*
- D. Chuẩn bị dữ liệu / *Data preparation*

3. Bài toán nào sau đây là phân cụm?

Which of the following problems is clustering?

- A. Dự báo dư nợ cuối tháng 10/2026 từ chuỗi 12 tháng trước / *Forecasting the month-end debt for October 2026 from the previous 12 months*
- B. Chia khách thành nhóm theo số giao dịch và chi tiêu khi chưa có nhóm cho trước / *Dividing customers into groups by transactions and spending when no groups are given*
- C. Đoán hồ sơ vay mới là Tốt hay Xấu từ các hồ sơ đã đáo hạn / *Predicting whether a new application is Good or Bad from matured applications*
- D. Tìm các cặp sản phẩm hay xuất hiện cùng nhau trong giao dịch / *Finding pairs of products that often appear together in transactions*

4. Trong 8 hồ sơ có 3 hồ sơ nợ xấu. Quy tắc luôn đoán "không nợ xấu" có độ chính xác bao nhiêu?

Among 8 applications 3 have bad debt. What is the accuracy of the rule that always predicts "no bad debt"?

- A. 37,5% / *37.5%*
- B. 50%
- C. 62,5% / *62.5%*
- D. 75%

5. Tập dữ liệu có cột nhãn nợ xấu đã biết; mục tiêu là đoán nhãn cho hồ sơ mới. Đây là kiểu học nào?
A data set has a known bad-debt label column; the goal is to predict the label of new applications. Which type of learning is this?
- A. Học không giám sát / *Unsupervised learning*
 - B. Học bán giám sát / *Semi-supervised learning*
 - C. Phân cụm / *Clustering*
 - D. Học có giám sát / *Supervised learning*
6. Mỗi ngày ứng dụng ghi 10.000 sự kiện, mỗi sự kiện 300 byte (1 MB = 1.000.000 byte). Dung lượng sau 30 ngày là
Each day the app records 10,000 events of 300 bytes each (1 MB = 1,000,000 bytes). The storage after 30 days is
- A. 9 MB
 - B. 30 MB
 - C. 90 MB
 - D. 900 MB
7. Mã Python "print(sum(x > 40 for x in [52, 46, 40, 30, 44]))" in ra giá trị nào?
Which value does the Python code "print(sum(x > 40 for x in [52, 46, 40, 30, 44]))" print?
- A. 2
 - B. 4
 - C. 3
 - D. 5
8. Theo Luật Bảo vệ dữ liệu cá nhân 91/2025 Điều 2 khoản 1, dữ liệu cá nhân sau khi khử nhận dạng thì
Under the Law on Personal Data Protection 91/2025, Article 2, clause 1, personal data after de-identification
- A. vẫn là dữ liệu cá nhân nhạy cảm / *remains sensitive personal data*
 - B. không còn là dữ liệu cá nhân / *is no longer personal data*
 - C. phải được xóa ngay lập tức / *must be deleted immediately*
 - D. trở thành bí mật nhà nước / *becomes a state secret*

Chương 2. Dữ liệu, thống kê mô tả và độ đo tương tự — Chapter 2. Data, descriptive statistics and similarity measures

Mục tiêu chương — Chapter objectives

- Phân loại thuộc tính theo kiểu (định danh, nhị phân, thứ bậc, số khoảng, số tỷ lệ) và theo tập giá trị (rời rạc, liên tục), rồi chọn phép toán hợp lệ cho từng kiểu.
Classify attributes by type (nominal, binary, ordinal, interval, ratio) and by value set (discrete, continuous), then choose the valid operations for each type.
- Tính tay trung bình, trung vị, mode, phương sai, độ lệch chuẩn, tứ phân vị, IQR và năm số tóm tắt của thu nhập khách hàng Phú Sa, phát hiện ngoại lai theo quy tắc $1.5 * IQR$.
Compute by hand the mean, median, mode, variance, standard deviation, quartiles, IQR and five-number summary of Phú Sa customer income, and detect outliers with the $1.5 * IQR$ rule.
- Đọc hình hộp, histogram và độ lệch; tính hệ số tương quan Pearson giữa thu nhập và dư nợ và nêu giới hạn của nó.
Read boxplots, histograms and skewness; compute the Pearson correlation between income and balance and state its limits.
- Lập ma trận khoảng cách sau chuẩn hóa min–max và tính khoảng cách Minkowski (Manhattan, Euclid, Chebyshev); kiểm tính chất metric.
Build a dissimilarity matrix after min–max normalisation and compute Minkowski distances (Manhattan, Euclid, Chebyshev); test the metric properties.
- Tính SMC, Jaccard, cosine và khoảng cách Gower cho dữ liệu nhị phân, định danh, thứ bậc và hỗn hợp.
Compute SMC, Jaccard, cosine and the Gower distance for binary, nominal, ordinal and mixed data.

2.1. Đối tượng, thuộc tính và kiểu thuộc tính — 2.1. Objects, attributes and attribute types

Một tập dữ liệu gồm các **đối tượng** (object; còn gọi là bản ghi, mẫu, điểm dữ liệu). Mỗi đối tượng được mô tả bằng một tập **thuộc tính** (attribute; còn gọi là biến, trường, cột, đặc trưng). Ở Công ty Tài chính TNHH Phú Sa, mỗi khách hàng là một đối tượng, gọi bằng mã KH01 đến KH12; mỗi khách được mô tả bằng bảy thuộc tính: tuổi, thu nhập, nghề, hạng thẻ, dư nợ, có nhà và số lần trễ hạn. Trong cơ sở dữ liệu quan hệ của học phần Cơ sở dữ liệu, đối tượng là một dòng (bộ), thuộc tính là một cột, và các hàm AVG, COUNT, MIN, MAX của SQL cho cùng kết quả với các độ đo ở mục 2.2 và 2.3.

*A data set consists of **objects** (also called records, samples or data points). Each object is described by a set of **attributes** (also called variables, fields, columns or features). At Phú Sa Finance Company Ltd, each customer is an object, named KH01 to KH12, and each customer is described by seven attributes: age, income, occupation, card tier, outstanding balance, home ownership and number of late payments. In the relational database of the Databases course an*

object is a row (tuple) and an attribute is a column, and the SQL functions AVG, COUNT, MIN and MAX give the same results as the measures in sections 2.2 and 2.3.

Mã ID	Tuổi Age	Thu nhập Income	Nghề Occupation	Hạng thẻ Card tier	Dư nợ Balance	Có nhà Home	Trễ hạn Late
KH01	25	9	Công nhân Factory worker	Chuẩn Standard	12	0	2
KH02	29	12	Nhân viên văn phòng Office clerk	Chuẩn Standard	18	0	1
KH03	34	15	Giáo viên Teacher	Vàng Gold	20	1	0
KH04	41	22	Kinh doanh tự do Self-employed	Vàng Gold	45	1	1
KH05	23	8	Công nhân Factory worker	Chuẩn Standard	10	0	4
KH06	38	18	Nhân viên văn phòng Office clerk	Vàng Gold	30	1	0
KH07	45	—	Kinh doanh tự do Self-employed	Vàng Gold	40	1	2
KH08	31	14	Giáo viên Teacher	Chuẩn Standard	16	0	0
KH09	27	11	Nhân viên văn phòng Office clerk	Chuẩn Standard	14	0	1
KH10	36	16	Công nhân Factory worker	Vàng Gold	25	1	3
KH11	50	20	Giáo viên Teacher	Bạch kim Platinum	35	1	0
KH12	47	120	Kinh doanh tự do Self-employed	Bạch kim Platinum	150	1	0

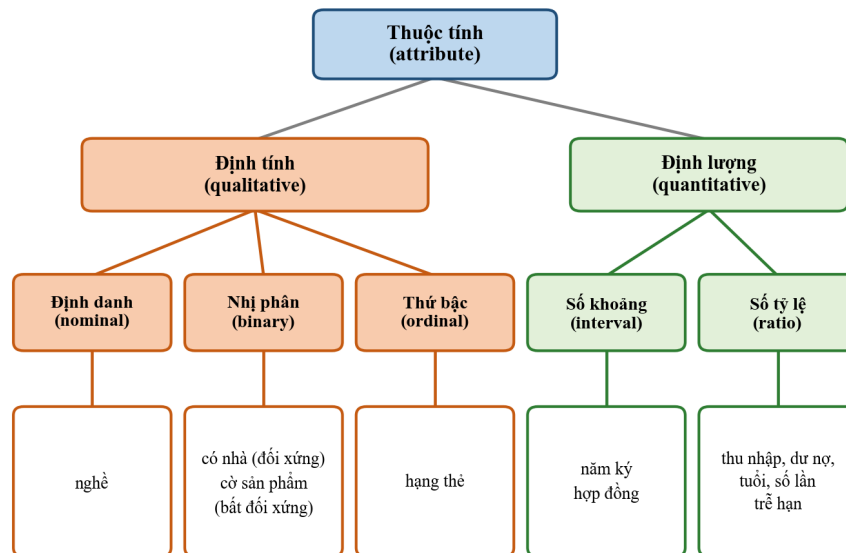
Thu nhập tính bằng triệu đồng mỗi tháng, dư nợ bằng triệu đồng; cột có nhà ghi 1 nếu khách sở hữu nhà, 0 nếu không. Ô thu nhập của KH07 bị thiếu (dấu gạch), nên các thống kê về thu nhập tính trên 11 khách; cách xử lý giá trị thiếu thuộc Chương 3. Khách KH12 có thu nhập 120 triệu đồng, gấp 8 lần trung vị, và sẽ là ngoại lai điển hình ở mục 2.3.

Income is in VND million per month and balance in VND million; the home column is 1 if the customer owns a home and 0 otherwise. The income cell of KH07 is missing (dash), so income statistics are computed over 11 customers; handling missing values belongs to Chapter 3. Customer KH12 earns VND 120 million, 8 times the median, and is the typical outlier of section 2.3.

Kiểu thuộc tính — *Attribute types*

Kiểu của thuộc tính quyết định phép toán nào có nghĩa: tính trung bình mã nghề là vô nghĩa, còn tính trung bình dư nợ thì có nghĩa. Có hai nhóm lớn, mỗi nhóm chia nhỏ như Hình 2.1: thuộc tính định tính (giá trị là nhãn) và thuộc tính định lượng (giá trị là số đo).

The type of an attribute decides which operations are meaningful: the average of an occupation code is meaningless, while the average balance is meaningful. There are two broad groups, subdivided as in Figure 2.1: qualitative attributes (values are labels) and quantitative attributes (values are measurements).



Ồ dưới mỗi kiểu: ví dụ lấy từ bảng khách hàng của Phú Sa

Hình 2.1. Phân loại thuộc tính, ví dụ lấy từ bảng khách hàng của Phú Sa
Figure 2.1. Classification of attributes, with examples from the Phú Sa customer table

Kiểu <i>Type</i>	Đặc điểm <i>Characteristics</i>	Phép có nghĩa <i>Meaningful operations</i>	Ví dụ ở Phú Sa <i>Phú Sa example</i>
Định danh <i>Nominal</i>	Giá trị là tên hoặc nhãn, không có thứ tự <i>Values are names or labels with no order</i>	So sánh bằng hoặc khác; đếm; mode <i>Equal or not equal; counting; mode</i>	Nghề: Giáo viên, Công nhân <i>Occupation: Teacher, Factory worker</i>
Nhị phân đối xứng <i>Symmetric binary</i>	Hai giá trị, hai giá trị có tầm quan trọng như nhau <i>Two values of equal importance</i>	Như định danh <i>As for nominal</i>	Có nhà: 1 hoặc 0 <i>Home ownership: 1 or 0</i>
Nhị phân bất đối xứng <i>Asymmetric binary</i>	Hai giá trị, giá trị 1 (có xuất hiện) mới mang thông tin <i>Two values, only 1 (present) carries information</i>	Như định danh; so sánh tập hợp (Jaccard) <i>As for nominal; set comparison (Jaccard)</i>	Cờ dùng bảo hiểm khoản vay <i>Flag for using loan insurance</i>

Kiểu <i>Type</i>	Đặc điểm <i>Characteristics</i>	Phép có nghĩa <i>Meaningful operations</i>	Ví dụ ở Phú Sa <i>Phú Sa example</i>
Thứ bậc <i>Ordinal</i>	Có thứ tự, khoảng cách giữa hai bậc không xác định <i>Ordered, gap between ranks undefined</i>	Thêm so sánh lớn hơn, nhỏ hơn; trung vị; tứ phân vị <i>Adds greater or less than; median; quartiles</i>	Hạng thẻ: Chuẩn < Vàng < Bạch kim <i>Card tier: Standard < Gold < Platinum</i>
Số khoảng <i>Interval</i>	Khoảng cách có nghĩa, không có điểm không tuyệt đối <i>Differences are meaningful, no true zero</i>	Thêm cộng, trừ; trung bình; độ lệch chuẩn; tỷ số vô nghĩa <i>Adds addition, subtraction, mean, standard deviation; ratios are meaningless</i>	Năm ký hợp đồng: 2019, 2024 <i>Contract year: 2019, 2024</i>
Số tỷ lệ <i>Ratio</i>	Có điểm không tuyệt đối, tỷ số có nghĩa <i>True zero, ratios are meaningful</i>	Mọi phép toán, kể cả tỷ số <i>All operations, including ratios</i>	Thu nhập 18 triệu gấp đôi 9 triệu <i>Income of 18 million is twice 9 million</i>

Cách phân chia khác theo tập giá trị: thuộc tính **rời rạc** (discrete) nhận hữu hạn hoặc đếm được giá trị, như số lần trễ hạn (0, 1, 2, 3, 4), hạng thẻ, nghề; thuộc tính **liên tục** (continuous) nhận giá trị thực trong một khoảng, như thu nhập, dư nợ. Thuộc tính nhị phân là trường hợp riêng của rời rạc. Dữ liệu máy tính chỉ lưu số có hữu hạn chữ số thập phân, nên liên tục là một mô hình của thuộc tính. Chương trình sau khai báo lại bảng khách hàng và đếm số giá trị khác nhau, số ô thiếu của từng thuộc tính.

*A second split is by the set of values: a **discrete** attribute takes finitely or countably many values, such as the number of late payments (0, 1, 2, 3, 4), card tier or occupation; a **continuous** attribute takes real values in an interval, such as income or balance. A binary attribute is a special case of a discrete one. A computer stores only numbers with finitely many decimals, so continuous is a model of the attribute. The following program declares the customer table again and counts the distinct values and the missing cells of each attribute.*

```
khach = [
    ("KH01", 25, 9, "Công nhân", "Chuẩn", 12, 0, 2),
    ("KH02", 29, 12, "Nhân viên văn phòng", "Chuẩn", 18, 0, 1),
    ("KH03", 34, 15, "Giáo viên", "Vàng", 20, 1, 0),
    ("KH04", 41, 22, "Kinh doanh tự do", "Vàng", 45, 1, 1),
    ("KH05", 23, 8, "Công nhân", "Chuẩn", 10, 0, 4),
    ("KH06", 38, 18, "Nhân viên văn phòng", "Vàng", 30, 1, 0),
    ("KH07", 45, None, "Kinh doanh tự do", "Vàng", 40, 1, 2),
    ("KH08", 31, 14, "Giáo viên", "Chuẩn", 16, 0, 0),
    ("KH09", 27, 11, "Nhân viên văn phòng", "Chuẩn", 14, 0, 1),
    ("KH10", 36, 16, "Công nhân", "Vàng", 25, 1, 3),
    ("KH11", 50, 20, "Giáo viên", "Bạch kim", 35, 1, 0),
    ("KH12", 47, 120, "Kinh doanh tự do", "Bạch kim", 150, 1, 0),
]
ten = ["ma", "tuoi", "thu_nhap", "nghe", "hang_the", "du_no",
       "co_nha", "so_lan_tru_han"]
for j, t in enumerate(ten):
    cot = [dong[j] for dong in khach]
    co_gia_tri = [x for x in cot if x is not None]
    print(f'{t:<15}{len(set(co_gia_tri)):>3} giá trị khác nhau, "
          f"thiếu {len(cot) - len(co_gia_tri)}")
```

Kết quả chạy / Output

ma	12 giá trị khác nhau, thiếu 0
tuoi	12 giá trị khác nhau, thiếu 0
thu_nhap	11 giá trị khác nhau, thiếu 1
nghe	4 giá trị khác nhau, thiếu 0
hang_the	3 giá trị khác nhau, thiếu 0
du_no	12 giá trị khác nhau, thiếu 0
co_nha	2 giá trị khác nhau, thiếu 0
so_lan_tre_han	5 giá trị khác nhau, thiếu 0

Thu nhập có 11 giá trị khác nhau và thiếu 1 ô; nghề có 4 giá trị, hạng thẻ có 3, số lần trễ hạn có 5. Mã khách là định danh với 12 giá trị khác nhau, nên không đưa vào tính toán.

Income has 11 distinct values and 1 missing cell; occupation has 4 values, card tier 3 and late payments 5. The customer code is a nominal identifier with 12 distinct values and is left out of calculations.

PHẢI THUỘC / MEMORISE — Định danh chỉ đếm; thứ bậc thêm trung vị; số khoảng thêm trung bình; số tỷ lệ thêm tỷ số. Nhị phân bất đối xứng dùng Jaccard, nhị phân đối xứng dùng khớp đơn giản (mục 2.7).

Nominal only counts; ordinal adds the median; interval adds the mean; ratio adds ratios. Asymmetric binary uses Jaccard, symmetric binary uses simple matching (section 2.7).

LỖI HAY GẶP / COMMON MISTAKE — Mã hóa hạng thẻ thành 1, 2, 3 rồi coi như số tỷ lệ: trung bình hạng 2,0 không có nghĩa là hạng Vàng, và hạng Bạch kim không phải gấp ba hạng Chuẩn. Cũng không tính trung bình mã khách hàng hay số điện thoại.

Coding card tier as 1, 2, 3 and treating it as a ratio attribute: an average rank of 2.0 does not mean Gold, and Platinum is not three times Standard. Do not average customer codes or phone numbers either.

2.2. Đo xu hướng trung tâm — 2.2. Measures of central tendency

Độ đo xu hướng trung tâm trả lời câu hỏi giá trị điển hình của thuộc tính là bao nhiêu. Với n giá trị $x_1, \dots, x_n$, sắp xếp tăng dần thành $x(1) \leq \dots \leq x(n)$, các độ đo gồm:

A measure of central tendency answers what the typical value of an attribute is. For n values $x_1, \dots, x_n$ sorted into $x(1) \leq \dots \leq x(n)$, the measures are:

$$\bar{x} = (\sum x_i) / n \quad (2.1)$$

- **Trung bình (mean)** cộng hết rồi chia n , công thức (2.1); dùng được cho số khoảng và số tỷ lệ.

The mean adds everything and divides by n , formula (2.1); usable for interval and ratio attributes.

$$\bar{x}_\alpha = (\sum x_{(i)}) / (n - 2k), i = k + 1, \dots, n - k; k = \lfloor \alpha * n \rfloor \quad (2.2)$$

- **Trung bình cắt tỉa (trimmed mean)** bỏ k giá trị nhỏ nhất và k giá trị lớn nhất rồi lấy trung bình, công thức (2.2). Sách này cắt tỉa 10% mỗi đầu ($\alpha = 0,1$).

The trimmed mean drops the k smallest and k largest values and averages the rest, formula (2.2). This book trims 10% at each end ($\alpha = 0.1$).

$$\text{trung vị} = x_{((n+1)/2)} \text{ khi } n \text{ lẻ; trung vị} = (x_{(n/2)} + x_{(n/2+1)}) / 2 \text{ khi } n \text{ chẵn} \quad (2.3)$$

- **Trung vị** (median) là giá trị ở giữa sau khi sắp xếp, công thức (2.3); dùng được từ thuộc tính thứ bậc trở lên.

*The **median** is the middle value after sorting, formula (2.3); usable from ordinal attributes upward.*

- **Mode** là giá trị xuất hiện nhiều nhất; có thể có nhiều mode hoặc không có mode nào; dùng được cho mọi kiểu thuộc tính.

*The **mode** is the most frequent value; there may be several modes or none; usable for every attribute type.*

$$\text{midrange} = (x_{\min} + x_{\max}) / 2 \quad (2.4)$$

- **Midrange** là trung điểm của giá trị nhỏ nhất và lớn nhất, công thức (2.4); tính nhanh nhưng bị ngoại lai chi phối.

*The **midrange** is the midpoint of the smallest and largest values, formula (2.4); quick to compute but dominated by outliers.*

```
import statistics as st
from collections import Counter

thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120] # bỏ KH07
tre_han = [2, 1, 0, 1, 4, 0, 2, 0, 1, 3, 0, 0] # cả 12 khách
n = len(thu_nhap)
x = sorted(thu_nhap)
k = int(0.1 * n) # cắt tía 10% mỗi đầu
print("n =", n, "| tổng =", sum(x))
print("trung bình =", round(st.mean(x), 2))
print("trung bình cắt tía 10% =", round(st.mean(x[k:n - k]), 2))
print("trung vị =", st.median(x))
print("midrange =", (x[0] + x[-1]) / 2)
print("thu nhập, giá trị gặp nhiều nhất:", Counter(x).most_common(1)[0])
mode, so_lan = Counter(tre_han).most_common(1)[0]
print("số lần trễ hạn: mode =", mode, "xuất hiện", so_lan, "lần")
```

Kết quả chạy / Output

```
n = 11 | tổng = 265
trung bình = 24.09
trung bình cắt tía 10% = 15.22
trung vị = 15
midrange = 64.0
thu nhập, giá trị gặp nhiều nhất: (8, 1)
số lần trễ hạn: mode = 0 xuất hiện 5 lần
```

Trung bình thu nhập là 24,09 triệu đồng, còn trung vị chỉ 15 triệu đồng: KH12 kéo trung bình lên 1,6 lần trung vị. Trung bình cắt tía 10% là 15,22, gần trung vị vì đã bỏ KH05 và KH12 (hai đầu). Midrange bằng 64 triệu đồng, không đại diện cho khách nào. Thu nhập không có mode vì mỗi giá trị chỉ xuất hiện một lần; số lần trễ hạn có mode 0 (5 khách). Với nghề, bốn nghề đều xuất hiện đúng 3 lần, nên tập dữ liệu có bốn mode; bộ đếm Counter của Python trả về Công nhân chỉ vì đó là nghề gặp đầu tiên trong danh sách.

Mean income is 24.09 million but the median is only 15 million: KH12 pulls the mean up to 1.6 times the median. The 10% trimmed mean is 15.22, close to the median because both ends (KH05 and KH12) are dropped. The midrange is 64 million and represents no customer. Income has no mode because every value occurs once; late payments have mode 0 (5 customers). For occupation all four occupations occur exactly 3 times, so the data set has four modes; the Python Counter returns Factory worker only because it is the first occupation met in the list.

PHẢI THUỘC / MEMORISE — Ngoại lai kéo trung bình và midrange lệch nhiều, còn trung vị gần như đứng yên. Phân phối đối xứng thì trung bình, trung vị trùng nhau; lệch phải thì trung bình lớn hơn trung vị.

Outliers drag the mean and the midrange far, and barely move the median. A symmetric distribution has mean equal to median; a right-skewed one has mean above median.

Bài mẫu 2.1. Trung bình, trung vị, cắt tỉa của dư nợ — *Worked example 2.1. Mean, median and trimmed mean of balance*

Đề bài: Bảng khách hàng có 12 giá trị dư nợ (triệu đồng): 12, 18, 20, 45, 10, 30, 40, 16, 14, 25, 35, 150. Tính trung bình, trung vị, trung bình cắt tỉa 10% và midrange.

Problem: the customer table has 12 balances (VND million): 12, 18, 20, 45, 10, 30, 40, 16, 14, 25, 35, 150. Compute the mean, the median, the 10% trimmed mean and the midrange.

Thứ tự Rank	Khách Customer	Dư nợ Balance	Thứ tự Rank	Khách Customer	Dư nợ Balance
1	KH05	10	7	KH10	25
2	KH01	12	8	KH06	30
3	KH09	14	9	KH11	35
4	KH08	16	10	KH07	40
5	KH02	18	11	KH04	45
6	KH03	20	12	KH12	150

- Trung bình: tổng = 415, $n = 12$, nên trung bình = $415 / 12 = 34,58$ triệu đồng.
Mean: sum = 415, $n = 12$, so the mean = $415 / 12 = 34.58$ million.
- Trung vị: $n = 12$ chẵn, lấy trung bình hai giá trị ở vị trí 6 và 7 sau khi sắp xếp: $(20 + 25) / 2 = 22,5$.
Median: $n = 12$ is even, so average the values at positions 6 and 7 after sorting: $(20 + 25) / 2 = 22.5$.
- Cắt tỉa 10%: $k = \lfloor 0,1 * 12 \rfloor = 1$, bỏ giá trị nhỏ nhất (10) và lớn nhất (150); trung bình 10 giá trị còn lại = $(415 - 10 - 150) / 10 = 25,5$.
10% trimmed mean: $k = \lfloor 0.1 * 12 \rfloor = 1$, drop the smallest (10) and the largest (150); mean of the remaining 10 values = $(415 - 10 - 150) / 10 = 25.5$.
- Midrange = $(10 + 150) / 2 = 80$.
Midrange = $(10 + 150) / 2 = 80$.
- Nhận xét: trung bình 34,58 lớn hơn trung vị 22,5 vì KH12 có dư nợ 150; bỏ hai đầu thì trung bình về 25,5. Dư nợ không có mode.
Comment: the mean 34.58 exceeds the median 22.5 because KH12 owes 150; dropping both ends brings the mean to 25.5. Balance has no mode.

2.3. Đo độ phân tán và năm số tóm tắt — 2.3. Measures of dispersion and the five-number summary

Hai tập thu nhập có cùng trung bình vẫn có thể rải rộng khác nhau, nên cần độ đo phân tán. Khoảng biến thiên (range) là hiệu giữa lớn nhất và nhỏ nhất. Phương sai đo trung bình bình phương độ lệch so với trung bình; độ lệch chuẩn là căn bậc hai của phương sai và có cùng đơn vị với dữ liệu.

Two income sets with the same mean can still be spread differently, so a measure of dispersion is needed. The range is the difference between the largest and smallest values. The variance is the mean squared deviation from the mean; the standard deviation is its square root and has the same unit as the data.

$$s^2 = \Sigma (x_i - \bar{x})^2 / (n - 1); \sigma^2 = \Sigma (x_i - \mu)^2 / n; s = \sqrt{s^2} \quad (2.5)$$

$$\Sigma (x_i - \bar{x})^2 = \Sigma x_i^2 - (\Sigma x_i)^2 / n \quad (2.6)$$

Đặc điểm <i>Feature</i>	Phương sai mẫu s^2 <i>Sample variance s^2</i>	Phương sai tổng thể σ^2 <i>Population variance σ^2</i>
Mẫu số <i>Denominator</i>	$n - 1$	n
Dùng khi <i>Use when</i>	Dữ liệu là mẫu lấy từ tổng thể lớn hơn <i>The data are a sample from a larger population</i>	Dữ liệu là toàn bộ tổng thể <i>The data are the whole population</i>
Hàm Python <i>Python function</i>	statistics.variance, statistics.stdev	statistics.pvariance, statistics pstdev
Mặc định của thư viện <i>Library defaults</i>	pandas (ddof = 1), Excel VAR.S	numpy.var (ddof = 0), Excel VAR.P

Sách này dùng phương sai và độ lệch chuẩn **mẫu** (mẫu số $n - 1$) cho mọi chương, trừ khi đề bài ghi rõ là tổng thể. Công thức (2.6) cho phép tính tay mà không cần từng độ lệch: chỉ cần tổng các giá trị và tổng bình phương.

*This book uses the **sample** variance and standard deviation (denominator $n - 1$) in every chapter, unless the problem says population. Formula (2.6) allows hand computation without each deviation: only the sum of values and the sum of squares are needed.*

```
import statistics as st

thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
n = len(thu_nhap)
tb = sum(thu_nhap) / n
tong_bp = sum((x - tb) ** 2 for x in thu_nhap)
print("khoảng biến thiên =", max(thu_nhap) - min(thu_nhap))
print("tổng bình phương lệch =", round(tong_bp, 2))
print("phương sai mẫu =", round(tong_bp / (n - 1), 2))
print("phương sai tổng thể =", round(tong_bp / n, 2))
print("độ lệch chuẩn mẫu =", round(st.stdev(thu_nhap), 2))
print("độ lệch chuẩn tổng thể =", round(st.pstdev(thu_nhap), 2))
print("hệ số biến thiên =", round(st.stdev(thu_nhap) / tb * 100, 1), "%")
```

Kết quả chạy / Output

```

khoảng biến thiên = 112
tổng bình phương lệch = 10310.91
phương sai mẫu = 1031.09
phương sai tổng thể = 937.36
độ lệch chuẩn mẫu = 32.11
độ lệch chuẩn tổng thể = 30.62
hệ số biến thiên = 133.3 %

```

Thu nhập có khoảng biến thiên 112, phương sai mẫu 1.031,09, độ lệch chuẩn mẫu 32,11 triệu đồng. Hệ số biến thiên (độ lệch chuẩn chia trung bình) là 133,3%, nghĩa là độ rải lớn hơn cả mức trung bình, một dấu hiệu của ngoại lai. Phương sai tổng thể nhỏ hơn (937,36) vì chia cho n .

Income has a range of 112, a sample variance of 1,031.09 and a sample standard deviation of 32.11 million. The coefficient of variation (standard deviation divided by the mean) is 133.3%, meaning the spread exceeds the mean itself, a sign of outliers. The population variance is smaller (937.36) because it divides by n .

Tứ phân vị, IQR và năm số tóm tắt — Quartiles, IQR and the five-number summary

Tứ phân vị chia dữ liệu đã sắp xếp thành bốn phần có số phần tử xấp xỉ bằng nhau: Q1 (25%), Q2 (trung vị), Q3 (75%). Hiệu IQR = Q3 – Q1 là độ rộng của nửa giữa dữ liệu. Năm số tóm tắt gồm giá trị nhỏ nhất, Q1, trung vị, Q3 và giá trị lớn nhất.

Quartiles split the sorted data into four parts with roughly equal counts: Q1 (25%), Q2 (median) and Q3 (75%). The difference IQR = Q3 – Q1 is the width of the middle half of the data. The five-number summary lists the minimum, Q1, the median, Q3 and the maximum.

CẢNH BÁO / WARNING — Quy ước tứ phân vị của sách (cố định cho mọi chương): vị trí của phân vị p là $1 + (n - 1) * p$ trong dãy đã sắp xếp, nội suy tuyến tính giữa hai giá trị kề; trùng với `statistics.quantiles(method="inclusive")` của Python và hàm `QUARTILE.INC` của Excel. Giáo trình khác có thể dùng quy ước khác và ra Q1, Q3 khác; đề thi cần ghi rõ quy ước.

Quartile convention of this book (fixed for all chapters): the position of percentile p is $1 + (n - 1) * p$ in the sorted list, with linear interpolation between neighbours; it matches Python `statistics.quantiles(method="inclusive")` and Excel `QUARTILE.INC`. Other textbooks may use another convention and obtain different Q1 and Q3; an exam should state the convention.

vị trí = $1 + (n - 1) * p$; $Q = x_{(j)} + f * (x_{(j+1)} - x_{(j)})$, với j là phần nguyên, f là phần thập phân của vị trí (2.7)

IQR = Q3 – Q1; hàng rào dưới = Q1 – 1,5 * IQR; hàng rào trên = Q3 + 1,5 * IQR (2.8)

Giá trị nằm ngoài hai hàng rào ở công thức (2.8) bị coi là ngoại lai theo quy tắc 1,5 * IQR của Tukey. Cũng có thể kiểm tra bằng z-score (bài toán nâng cao ở Chương 9).

*A value outside the two fences of formula (2.8) is treated as an outlier under Tukey's 1.5 * IQR rule. A z-score check is also possible (an advanced topic in Chapter 9).*

```
import statistics as st

thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
x = sorted(thu_nhap)
q1, q2, q3 = st.quantiles(x, n=4, method="inclusive")
iqr = q3 - q1
thap, cao = q1 - 1.5 * iqr, q3 + 1.5 * iqr
print("năm số:", x[0], q1, q2, q3, x[-1])
print("IQR =", iqr, "| hàng rào:", thap, cao)
ngoai_lai = [v for v in x if v < thap or v > cao]
print("ngoại lai:", ngoai_lai)
con_lai = [v for v in x if v not in ngoai_lai]
print("trung bình bỏ ngoại lai =", round(st.mean(con_lai), 2))
print("quy ước exclusive:", st.quantiles(x, n=4, method="exclusive"))
```

Kết quả chạy / Output

```
năm số: 8 11.5 15.0 19.0 120
IQR = 7.5 | hàng rào: 0.25 30.25
ngoại lai: [120]
trung bình bỏ ngoại lai = 14.5
quy ước exclusive: [11.0, 15.0, 20.0]
```

Năm số tóm tắt của thu nhập là 8; 11,5; 15; 19; 120. IQR = 7,5; hàng rào là 0,25 và 30,25, nên chỉ KH12 (120) là ngoại lai. Bỏ KH12, thu nhập trung bình còn 14,5 triệu đồng, sát trung vị 15. Với quy ước exclusive (mẫu số $n + 1$), Q1 = 11 và Q3 = 20, IQR = 9: kết luận ngoại lai vẫn như cũ nhưng hàng rào đổi, vì vậy phải ghi rõ quy ước.

The five-number summary of income is 8, 11.5, 15, 19, 120. IQR = 7.5; the fences are 0.25 and 30.25, so only KH12 (120) is an outlier. Without KH12 the mean income falls to 14.5 million, close to the median 15. Under the exclusive convention (denominator $n + 1$), Q1 = 11 and Q3 = 20, IQR = 9: the outlier verdict is unchanged but the fences move, which is why the convention must be stated.

Bài mẫu 2.2. Phương sai và độ lệch chuẩn của thu nhập — *Worked example 2.2.*

Variance and standard deviation of income

Đề bài: Tính phương sai mẫu và độ lệch chuẩn mẫu của 11 giá trị thu nhập (bỏ KH07), dùng công thức (2.6).

Problem: compute the sample variance and sample standard deviation of the 11 income values (KH07 excluded), using formula (2.6).

Khách <i>Customer</i>	xi	xi ²	Khách <i>Customer</i>	xi	xi ²
KH01	9	81	KH08	14	196
KH02	12	144	KH09	11	121
KH03	15	225	KH10	16	256
KH04	22	484	KH11	20	400
KH05	8	64	KH12	120	14400
KH06	18	324	Cộng <i>Sum</i>	265	16.695

1. Từ bảng: $n = 11$, $\sum x_i = 265$, $\sum x_i^2 = 16.695$.
From the table: $n = 11$, $\sum x_i = 265$, $\sum x_i^2 = 16,695$.
2. Tổng bình phương lệch: $\sum x_i^2 - (\sum x_i)^2 / n = 16.695 - 265^2 / 11 = 16.695 - 6.384,09 = 10.310,91$.
Sum of squared deviations: $\sum x_i^2 - (\sum x_i)^2 / n = 16,695 - 265^2 / 11 = 16,695 - 6,384.09 = 10,310.91$.
3. Phương sai mẫu: $s^2 = 10.310,91 / (11 - 1) = 1.031,09$.
Sample variance: $s^2 = 10,310.91 / (11 - 1) = 1,031.09$.
4. Độ lệch chuẩn mẫu: $s = \sqrt{1.031,09} = 32,11$ triệu đồng.
Sample standard deviation: $s = \sqrt{1,031.09} = 32.11$ million.

Bài mẫu 2.3. Năm số tóm tắt và ngoại lai của thu nhập — *Worked example 2.3. Five-number summary and outliers of income*

Đề bài: Tìm năm số tóm tắt, IQR và các ngoại lai của 11 giá trị thu nhập theo quy ước của sách.

Problem: find the five-number summary, the IQR and the outliers of the 11 income values under the book's convention.

Vị trí Position	1	2	3	4	5	6	7	8	9	10	11
Giá trị Value	8	9	11	12	14	15	16	18	20	22	120

1. Sắp xếp tăng dần (bảng trên). Giá trị nhỏ nhất 8, lớn nhất 120.
Sort ascending (table above). The minimum is 8 and the maximum is 120.
2. Q1: vị trí = $1 + 10 * 0,25 = 3,5$, giữa giá trị thứ 3 (11) và thứ 4 (12): $Q1 = 11 + 0,5 * (12 - 11) = 11,5$.
Q1: position = $1 + 10 * 0.25 = 3.5$, between the 3rd (11) and 4th (12) values: $Q1 = 11 + 0.5 * (12 - 11) = 11.5$.
3. Trung vị: vị trí = $1 + 10 * 0,5 = 6$, giá trị thứ 6 là 15.
Median: position = $1 + 10 * 0.5 = 6$, and the 6th value is 15.
4. Q3: vị trí = $1 + 10 * 0,75 = 8,5$, giữa giá trị thứ 8 (18) và thứ 9 (20): $Q3 = 18 + 0,5 * (20 - 18) = 19$.
Q3: position = $1 + 10 * 0.75 = 8.5$, between the 8th (18) and 9th (20) values: $Q3 = 18 + 0.5 * (20 - 18) = 19$.
5. $IQR = 19 - 11,5 = 7,5$; $1,5 * IQR = 11,25$. Hàng rào dưới = $11,5 - 11,25 = 0,25$; hàng rào trên = $19 + 11,25 = 30,25$.
 $IQR = 19 - 11.5 = 7.5$; $1.5 * IQR = 11.25$. Lower fence = $11.5 - 11.25 = 0.25$; upper fence = $19 + 11.25 = 30.25$.
6. Giá trị lớn nhất nằm trong hàng rào là 22; giá trị 120 vượt 30,25 nên là ngoại lai duy nhất.
The largest value inside the fences is 22; the value 120 exceeds 30.25 and is the only outlier.

PHẢI THUỘC / MEMORISE — Phương sai mẫu chia $n - 1$, tổng thể chia n . Tứ phân vị của sách: vị trí $1 + (n - 1) * p$. Ngoại lai: ngoài $[Q1 - 1,5 * IQR; Q3 + 1,5 * IQR]$.

Sample variance divides by $n - 1$, population variance by n . Book quartiles: position $1 + (n - 1) * p$. Outlier: outside $[Q1 - 1.5 * IQR; Q3 + 1.5 * IQR]$.

LỖI HAY GẶP / COMMON MISTAKE — Chia cho n khi đề cho mẫu (hoặc $n - 1$ khi đề cho cả tổng thể); tự tính Q1 theo cách khác quy ước đề; quên bình phương trong công thức phương sai; lấy căn rồi mới chia.

Dividing by n when the problem gives a sample (or $n - 1$ for a whole population); computing $Q1$ by a different convention from the problem; forgetting the square in the variance; taking the root before dividing.

2.4. Hình hộp, histogram và độ lệch — 2.4. Boxplots, histograms and skewness

Histogram chia trục giá trị thành các khoảng liên tiếp cùng độ rộng và vẽ cột cao bằng số quan sát trong mỗi khoảng. **Hình hộp** (boxplot) vẽ năm số tóm tắt: hộp đi từ Q1 đến Q3, vạch trong hộp là trung vị, hai râu kéo tới giá trị xa nhất còn nằm trong hàng rào, các điểm ngoài hàng rào vẽ riêng. Hình 2.2 vẽ hai biểu đồ của thu nhập trên cùng trục.

A histogram splits the value axis into consecutive intervals of equal width and draws bars whose height is the number of observations in each interval. A boxplot draws the five-number summary: the box runs from $Q1$ to $Q3$, the line inside is the median, the two whiskers reach the farthest value still inside the fences, and points outside the fences are drawn separately. Figure 2.2 draws both charts of income on the same axis.

```
import statistics as st

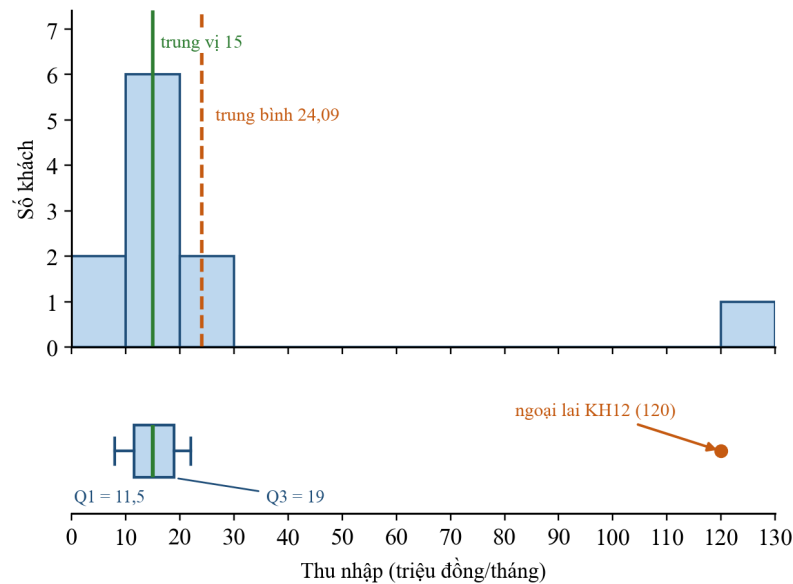
thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
dem = {}
for v in thu_nhap:
    khoang = v // 10 * 10
    dem[khoang] = dem.get(khoang, 0) + 1
for a in range(0, 130, 10):
    so = dem.get(a, 0)
    print(f"[{a}>{a+10}] {'#' * so} {so}")
n = len(thu_nhap)
tb, s = st.mean(thu_nhap), st.stdev(thu_nhap)
m2 = sum((x - tb) ** 2 for x in thu_nhap) / n
m3 = sum((x - tb) ** 3 for x in thu_nhap) / n
print("g1 =", round(m3 / m2 ** 1.5, 2))
print("3(tb - trung vị)/s =", round(3 * (tb - st.median(thu_nhap)) / s, 2))
```

Kết quả chạy / Output

```
[ 0, 10) ## 2
[ 10, 20) ##### 6
[ 20, 30) ## 2
[ 30, 40) 0
[ 40, 50) 0
[ 50, 60) 0
[ 60, 70) 0
[ 70, 80) 0
[ 80, 90) 0
[ 90, 100) 0
[100, 110) 0
[110, 120) 0
[120, 130) # 1
```

$$g_1 = 2.75$$

$$3(tb - \text{trung vị})/s = 0.85$$



Hình 2.2. Histogram (độ rộng 10) và hình hộp của thu nhập 11 khách, đơn vị triệu đồng/tháng
Figure 2.2. Histogram (bin width 10) and boxplot of the income of 11 customers, VND million per month

Hộp đi từ 11,5 đến 19 với vạch trung vị ở 15; râu trái tới 8, râu phải tới 22 (giá trị lớn nhất còn trong hàng rào 30,25); KH12 là điểm tách riêng. Histogram cho thấy 10 khách dồn trong khoảng 8 đến 22 và một cột đơn độc ở [120, 130).

The box runs from 11.5 to 19 with the median line at 15; the left whisker reaches 8 and the right whisker reaches 22 (the largest value inside the fence 30.25); KH12 is the separate point. The histogram shows 10 customers packed between 8 and 22 and a lone bar at [120, 130).

Độ lệch (skewness) — Skewness

Độ lệch mô tả phân phối cân đối hay không. Phân phối **lệch phải** (dương) có đuôi dài về phía giá trị lớn và trung bình lớn hơn trung vị; **lệch trái** (âm) ngược lại; **đối xứng** thì hai độ đo gần bằng nhau. Hai hệ số hay dùng là hệ số moment g_1 và hệ số thứ hai của Pearson:

*Skewness describes whether a distribution is balanced. A **right-skewed** (positive) distribution has a long tail towards large values and a mean above the median; **left-skewed** (negative) is the opposite; a **symmetric** one has the two nearly equal. Two common coefficients are the moment coefficient g_1 and Pearson's second coefficient:*

$$g_1 = [\Sigma (x_i - \bar{x})^3 / n] / [\Sigma (x_i - \bar{x})^2 / n]^{3/2} \quad (2.9)$$

$$Sk = 3 * (\bar{x} - \text{trung vị}) / s \quad (2.10)$$

Thu nhập có $g_1 = 2,75$ và hệ số thứ hai của Pearson là 0,85. Cả hai dương, khớp với đồ thị: phân phối lệch phải mạnh do KH12. Giá trị g_1 lớn hơn 1 thường được xem là lệch mạnh, nhưng với 11 quan sát thì hệ số này bất ổn và chỉ nên dùng để mô tả.

Income has $g_1 = 2.75$ and Pearson's second coefficient 0.85. Both are positive, matching the chart: a strongly right-skewed distribution caused by KH12. A g_1 above 1 is usually read as

strong skew, but with 11 observations the coefficient is unstable and should only be used descriptively.

PHẢI THUỘC / MEMORISE — Lệch phải: trung bình > trung vị, đuôi dài bên phải, hộp lệch về phía nhỏ. Lệch trái ngược lại. Hình hộp cho thấy ngoại lai; histogram cho thấy hình dạng.

Right skew: mean > median, long right tail, box shifted to the small side. Left skew is the opposite. A boxplot shows outliers; a histogram shows shape.

LỖI HAY GẶP / COMMON MISTAKE — Đổi độ rộng khoảng của histogram làm hình dạng thay đổi (10 so với 5 hay 20); kết luận về phân phối từ histogram cần nêu độ rộng đã chọn. Với 11 điểm, đừng khẳng định phân phối chuẩn hay lệch dựa trên hình.

Changing the bin width changes the shape (10 versus 5 or 20); a conclusion drawn from a histogram must state the width chosen. With 11 points, do not assert normality or skew from the picture alone.

2.5. Tương quan Pearson và biểu đồ phân tán — 2.5. Pearson correlation and the scatter plot

Biểu đồ phân tán (scatter plot) đặt mỗi đối tượng thành một điểm theo hai thuộc tính số. **Hệ số tương quan Pearson** r tóm tắt mức tuyến tính của đám điểm bằng một số trong đoạn $[-1, 1]$: r gần 1 là cùng tăng, gần -1 là một tăng một giảm, gần 0 là không có quan hệ tuyến tính. Bình phương r^2 là phần biến thiên của thuộc tính này được thuộc tính kia giải thích bằng đường thẳng.

*A scatter plot places each object as a point by two numeric attributes. The **Pearson correlation coefficient** r summarises how linear the cloud is as a number in $[-1, 1]$: r near 1 means rising together, near -1 means one rises as the other falls, near 0 means no linear relation. The square r^2 is the share of variation in one attribute explained by the other through a straight line.*

$$r = \Sigma (x_i - \bar{x})(y_i - \bar{y}) / \sqrt{\Sigma (x_i - \bar{x})^2 * \Sigma (y_i - \bar{y})^2} \quad (2.11)$$

```
import math

thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
du_no = [12, 18, 20, 45, 10, 30, 16, 14, 25, 35, 150]

def pearson(x, y):
    mx, my = sum(x) / len(x), sum(y) / len(y)
    t = sum((a - mx) * (b - my) for a, b in zip(x, y))
    m = sum((a - mx) ** 2 for a in x) * sum((b - my) ** 2 for b in y)
    return t / math.sqrt(m)

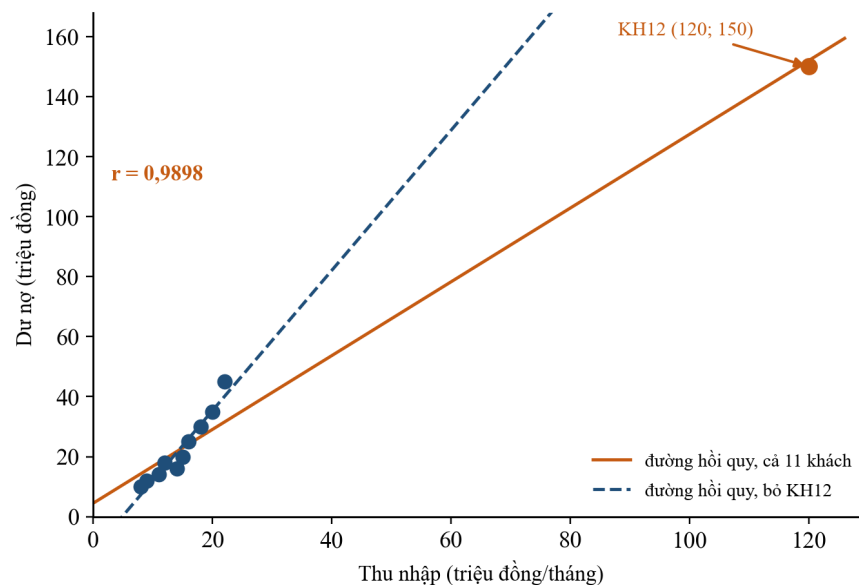
r = pearson(thu_nhap, du_no)
print("r (11 khách) =", round(r, 4), "| r bình phương =", round(r * r, 4))
giu = [i for i, v in enumerate(thu_nhap) if v < 100] # bỏ KH12
x = [thu_nhap[i] for i in giu]
y = [du_no[i] for i in giu]
print("r (bỏ KH12) =", round(pearson(x, y), 4))
```

Kết quả chạy / Output

```
r (11 khách) = 0.9898 | r bình phương = 0.9797
r (bỏ KH12) = 0.9627
```

Trên 11 khách có đủ hai giá trị, $r = 0,9898$ và $r^2 = 0,9797$. KH12 nằm xa đám còn lại nhưng cùng hướng nên kéo r lên; bỏ KH12 thì $r = 0,9627$. Cả hai đều sát 1 (cùng tăng), nhưng đường hồi quy ở Hình 2.3 thay đổi độ dốc nhiều khi bỏ hoặc giữ một điểm, tức hệ số Pearson nhạy với ngoại lai.

Over the 11 customers with both values, $r = 0.9898$ and $r^2 = 0.9797$. KH12 lies far from the rest but in the same direction, so it lifts r ; without KH12, $r = 0.9627$. Both are close to 1 (rising together), yet the regression line in Figure 2.3 changes its slope a lot when one point is kept or dropped, so the Pearson coefficient is sensitive to outliers.



Hình 2.3. Biểu đồ phân tán thu nhập và dư nợ của 11 khách (KH07 thiếu thu nhập nên bị loại)
Figure 2.3. Scatter plot of income and balance of 11 customers (KH07 has no income and is excluded)

Hai giới hạn cần nhớ. Một là Pearson chỉ đo quan hệ tuyến tính: một đường cong chữ U có thể cho r gần 0. Hai là tương quan không chứng minh nhân quả: dư nợ lớn đi cùng thu nhập cao vì chính sách hạn mức của công ty phụ thuộc thu nhập, không phải vì dư nợ làm thu nhập tăng. Quan hệ giữa hai thuộc tính định danh được kiểm bằng χ^2 (Chương 3).

Two limits to remember. First, Pearson measures only linear relations: a U-shaped curve can give r near 0. Second, correlation does not prove causation: a large balance goes with a high income because the company's credit limits depend on income, not because the balance raises income. The relation between two nominal attributes is tested with χ^2 (Chapter 3).

Bài mẫu 2.4. Hệ số Pearson trên 5 khách — Worked example 2.4. Pearson coefficient over 5 customers

Đề bài: Tính r giữa thu nhập và dư nợ của KH01, KH03, KH05, KH06, KH11 (bảng khách hàng, mục 2.1).

Problem: compute r between income and balance for KH01, KH03, KH05, KH06 and KH11 (customer table, section 2.1).

Khách <i>Customer</i>	x	y	$x - \bar{x}$	$y - \bar{y}$	$(x - \bar{x})(y - \bar{y})$	$(x - \bar{x})^2$	$(y - \bar{y})^2$
KH01	9	12	-5,0	-9,4	47,0	25,0	88,36
KH03	15	20	1,0	-1,4	-1,4	1,0	1,96
KH05	8	10	-6,0	-11,4	68,4	36,0	129,96
KH06	18	30	4,0	8,6	34,4	16,0	73,96
KH11	20	35	6,0	13,6	81,6	36,0	184,96
Cộng <i>Sum</i>	70	107	0	0	230,0	114,0	479,20

1. Trung bình: $\bar{x} = 70 / 5 = 14$; $\bar{y} = 107 / 5 = 21,4$.

Means: $\bar{x} = 70 / 5 = 14$; $\bar{y} = 107 / 5 = 21,4$.

2. Điền bảng: tổng tích độ lệch = 230,0; tổng bình phương độ lệch của x = 114,0, của y = 479,20.

Fill the table: sum of products of deviations = 230.0; sum of squared deviations of x = 114.0, of y = 479.20.

3. $r = 230,0 / \sqrt{(114,0 * 479,20)} = 230,0 / 233,73 = 0,9840$.

$r = 230.0 / \sqrt{(114.0 * 479.20)} = 230.0 / 233.73 = 0.9840$.

4. Kết luận: r gần 1, hai thuộc tính cùng tăng theo quan hệ tuyến tính mạnh trên 5 khách này.

Conclusion: r is close to 1; the two attributes rise together in a strong linear relation over these 5 customers.

LỖI HAY GẶP / COMMON MISTAKE — Kết luận nhân quả từ r cao; bỏ qua ngoại lai khi báo r; tính r trên các cặp không cùng khách (lệch thứ tự dòng, hoặc quên loại KH07 khi thu nhập thiếu).

Concluding causation from a high r; ignoring outliers when reporting r; computing r over pairs that are not the same customer (shifted row order, or forgetting to drop KH07 when income is missing).

2.6. Ma trận dữ liệu, ma trận khoảng cách và khoảng cách Minkowski — 2.6.

Data matrix, dissimilarity matrix and Minkowski distance

Các thuật toán phân cụm, kNN và phát hiện bất thường (Chương 6, 8, 9) cần một con số đo hai đối tượng giống hay khác nhau. **Ma trận dữ liệu** X có n dòng (đối tượng) và p cột (thuộc tính). **Ma trận khoảng cách** D là ma trận vuông $n \times n$, phần tử d_{ij} là khoảng cách giữa đối tượng i và j; D đối xứng và có đường chéo bằng 0. Ví dụ với bốn khách và ba thuộc tính số (tuổi, thu nhập, số lần trễ hạn):

*Clustering, kNN and anomaly detection (Chapters 6, 8, 9) need a number measuring how alike or different two objects are. The **data matrix** X has n rows (objects) and p columns (attributes). The **dissimilarity matrix** D is an $n \times n$ square matrix whose entry d_{ij} is the distance between objects i and j; D is symmetric with a zero diagonal. Example with four customers and three numeric attributes (age, income, late payments):*

$$X = \begin{bmatrix} 34 & 15 & 0 \\ 38 & 18 & 0 \\ 31 & 14 & 0 \\ 36 & 16 & 3 \end{bmatrix} \quad \begin{matrix} \text{dòng: KH03, KH06, KH08,} \\ \text{KH10} \end{matrix}$$

Ba thuộc tính có thang đo khác nhau: thu nhập trải từ 8 đến 120, số lần trễ hạn chỉ từ 0 đến 4. Nếu tính khoảng cách trên số gốc, thu nhập chi phối và số lần trễ hạn gần như vô hình. Cách xử lý thông dụng là chuẩn hóa min–max đưa mọi thuộc tính về $[0, 1]$ (chi tiết ở Chương 3), với min và max lấy trên toàn bộ cột của 12 khách:

The three attributes have different scales: income spans 8 to 120, late payments only 0 to 4. On raw numbers income dominates and late payments become almost invisible. The usual remedy is min–max normalisation, mapping every attribute to $[0, 1]$ (details in Chapter 3), with min and max taken over the whole column of the 12 customers:

$$x' = (x - \min) / (\max - \min) \quad (2.12)$$

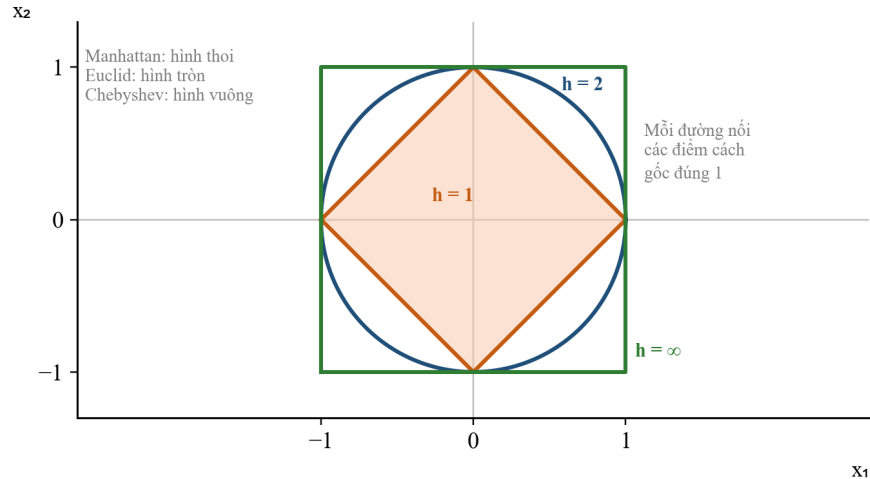
Khoảng cách **Minkowski** bậc h giữa hai đối tượng x và y có p thuộc tính là:

*The **Minkowski** distance of order h between two objects x and y with p attributes is:*

$$d(x, y) = (\sum |x_f - y_f|^h)^{1/h}; \quad h = 1: \text{Manhattan}; \quad h = 2: \text{Euclid}; \quad h = \infty: \text{Chebyshev} = \max |x_f - y_f| \quad (2.13)$$

Trong công thức (2.13) chữ h là bậc (nhiều giáo trình gọi là p , trùng với số thuộc tính nên sách đổi sang h để khỏi nhầm). Bậc càng lớn thì khoảng cách càng bị thành phần lệch nhiều nhất chi phối; khi h tăng, d không tăng và tiến tới Chebyshev. Hình 2.4 vẽ tập các điểm cách gốc đúng 1 theo ba bậc.

In formula (2.13) the letter h is the order (many textbooks call it p , which clashes with the number of attributes, so the book uses h). The larger the order, the more the distance is dominated by the component that differs most; as h grows, d does not increase and tends to the Chebyshev distance. Figure 2.4 draws the set of points at distance exactly 1 from the origin for three orders.



Hình 2.4. Tập điểm cách gốc đúng 1 theo khoảng cách Manhattan, Euclid, Chebyshev
Figure 2.4. Points at distance exactly 1 from the origin under the Manhattan, Euclid and Chebyshev distances

Khoảng cách <i>Distance</i>	Công thức <i>Formula</i>	Đặc điểm <i>Characteristics</i>	Khi nên dùng <i>When to use</i>
Manhattan ($h = 1$)	$\sum x_f - y_f $	Cộng các chênh lệch; ít nhạy ngoại lai hơn Euclid <i>Adds the differences; less sensitive to outliers than Euclid</i>	Dữ liệu nhiều chiều, thuộc tính rời rạc <i>High-dimensional data, discrete attributes</i>
Euclid ($h = 2$)	$\sqrt{(\sum (x_f - y_f)^2)}$	Độ dài đường thẳng; thuật toán k-means dựa trên nó <i>Straight-line length; k-means is built on it</i>	Mặc định cho thuộc tính số đã chuẩn hóa <i>Default for normalised numeric attributes</i>
Chebyshev ($h = \infty$)	$\max x_f - y_f $	Chỉ lấy chênh lệch lớn nhất <i>Takes only the largest difference</i>	Khi một thuộc tính lệch quá ngưỡng là đủ coi hai đối tượng khác nhau <i>When one attribute exceeding a limit is enough to call two objects different</i>

```

lo = (23, 8, 0)          # nhỏ nhất của tuổi, thu nhập, số lần trễ hạn
hi = (50, 120, 4)       # lớn nhất, lấy trên cả 12 khách
khach = {"KH06": (38, 18, 0), "KH10": (36, 16, 3)}

def chuan(v):
    return [(x - a) / (b - a) for x, a, b in zip(v, lo, hi)]

a, b = chuan(khach["KH06"]), chuan(khach["KH10"])
print("KH06:", [round(x, 4) for x in a])
print("KH10:", [round(x, 4) for x in b])
hieus = [abs(x - y) for x, y in zip(a, b)]
print("Manhattan =", round(sum(hieus), 4))
print("Euclid     =", round(sum(h ** 2 for h in hieus) ** 0.5, 4))
    
```

```
print("Minkowski3 =", round(sum(h ** 3 for h in hieu) ** (1 / 3), 4))
print("Chebyshev =", round(max(hieu), 4))
```

Kết quả chạy / Output

```
KH06: [0.5556, 0.0893, 0.0]
KH10: [0.4815, 0.0714, 0.75]
Manhattan = 0.8419
Euclid = 0.7539
Minkowski3 = 0.7502
Chebyshev = 0.75
```

Giữa KH06 và KH10: Manhattan 0,8419, Euclid 0,7539, Minkowski bậc 3 0,7502, Chebyshev 0,7500. Thứ tự giảm dần theo bậc, và Chebyshev trùng với chênh lệch của số lần trễ hạn (3 lần so với 0 lần, đã chuẩn hóa thành 0,75), thành phần lệch nhiều nhất.

Between KH06 and KH10: Manhattan 0.8419, Euclid 0.7539, Minkowski of order 3 0.7502, Chebyshev 0.7500. The values fall as the order rises, and Chebyshev equals the difference in late payments (3 versus 0, normalised to 0.75), the component that differs most.

Tính chất metric — Metric properties

Một hàm d là **metric** (khoảng cách theo nghĩa toán học) khi thỏa bốn tính chất với mọi đối tượng x, y, z :

*A function d is a **metric** (a distance in the mathematical sense) when it satisfies four properties for all objects x, y, z :*

- Không âm: $d(x, y) \geq 0$.
Non-negativity: $d(x, y) \geq 0$.
- Đồng nhất: $d(x, y) = 0$ khi và chỉ khi $x = y$.
Identity: $d(x, y) = 0$ if and only if $x = y$.
- Đối xứng: $d(x, y) = d(y, x)$.
Symmetry: $d(x, y) = d(y, x)$.
- Bất đẳng thức tam giác: $d(x, z) \leq d(x, y) + d(y, z)$.
Triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$.

Manhattan, Euclid, Chebyshev và Minkowski bậc từ 1 trở lên đều là metric. Bậc nhỏ hơn 1 vi phạm bất đẳng thức tam giác; $1 - \cos$ cũng không thỏa tính chất này nói chung. Chương trình sau dựng ma trận khoảng cách Euclid của bốn khách ở ma trận X và thử bất đẳng thức tam giác trên mọi bộ ba.

Manhattan, Euclid, Chebyshev and Minkowski of order 1 or more are all metrics. An order less than 1 violates the triangle inequality; $1 - \cos$ also fails it in general. The following program builds the Euclid dissimilarity matrix of the four customers in matrix X and tests the triangle inequality on every triple.

```
from itertools import permutations

lo, hi = (23, 8, 0), (50, 120, 4)
khach = {"KH03": (34, 15, 0), "KH06": (38, 18, 0),
         "KH08": (31, 14, 0), "KH10": (36, 16, 3)}

def chuan(v):
    return [(x - a) / (b - a) for x, a, b in zip(v, lo, hi)]
```

```
def d(u, v):
    a, b = chuan(khach[u]), chuan(khach[v])
    return sum((x - y) ** 2 for x, y in zip(a, b)) ** 0.5

ten = list(khach)
print("      " + "".join(f"{t:>8}" for t in ten))
for t in ten:
    print(f"{t:<6}" + "".join(f"{d(t, u):8.4f}" for u in ten))
dung = all(d(a, c) <= d(a, b) + d(b, c) + 1e-12
           for a, b, c in permutations(ten, 3))
print("bất đẳng thức tam giác đúng với mọi bộ ba:", dung)
```

Kết quả chạy / Output

	KH03	KH06	KH08	KH10
KH03	0.0000	0.1506	0.1115	0.7537
KH06	0.1506	0.0000	0.2617	0.7539
KH08	0.1115	0.2617	0.0000	0.7727
KH10	0.7537	0.7539	0.7727	0.0000

bất đẳng thức tam giác đúng với mọi bộ ba: True

$$D = \begin{bmatrix} 0,0000 & 0,1506 & 0,1115 & 0,7537 \\ 0,1506 & 0,0000 & 0,2617 & 0,7539 \\ 0,1115 & 0,2617 & 0,0000 & 0,7727 \\ 0,7537 & 0,7539 & 0,7727 & 0,0000 \end{bmatrix} \quad \begin{matrix} \text{dòng, cột: KH03, KH06,} \\ \text{KH08, KH10} \end{matrix}$$

Ma trận D đối xứng, đường chéo bằng 0, và phần tử giữa KH06 và KH10 là 0,7539, khớp với kết quả ở trên. KH08 gần KH03 nhất (0,1115), còn KH10 cách xa ba khách kia vì số lần trễ hạn là 3 trong khi ba khách còn lại là 0.

Matrix D is symmetric with a zero diagonal, and the entry between KH06 and KH10 is 0.7539, matching the result above. KH08 is closest to KH03 (0.1115), while KH10 is far from the other three because its late payments are 3 against 0 for the rest.

Bài mẫu 2.5. Khoảng cách giữa KH06 và KH10 sau khi chuẩn hóa — *Worked example 2.5. Distance between KH06 and KH10 after normalisation*

Đề bài: Tính khoảng cách Manhattan, Euclid và Chebyshev giữa KH06 và KH10 theo ba thuộc tính tuổi, thu nhập, số lần trễ hạn, sau khi chuẩn hóa min–max trên cả 12 khách.

Problem: compute the Manhattan, Euclid and Chebyshev distances between KH06 and KH10 over age, income and late payments, after min–max normalisation over all 12 customers.

Thuộc tính <i>Attribute</i>	Min	Max	KH0 6	KH06 chuẩn hóa <i>KH06 scaled</i>	KH1 0	KH10 chuẩn hóa <i>KH10 scaled</i>	Hiệu <i>Difference</i>
Tuổi <i>Age</i>	23	50	38	0,5556	36	0,4815	0,0741
Thu nhập <i>Income</i>	8	120	18	0,0893	16	0,0714	0,0179
Số lần trễ hạn <i>Late payments</i>	0	4	0	0,0000	3	0,7500	0,7500

- Chuẩn hóa từng giá trị theo $(x - \min) / (\max - \min)$, ví dụ tuổi KH06: $(38 - 23) / (50 - 23) = 15 / 27 = 0,5556$; thu nhập KH10: $(16 - 8) / (120 - 8) = 8 / 112 = 0,0714$; số lần trễ hạn KH10: $3 / 4 = 0,75$.

Normalise each value with $(x - \min) / (\max - \min)$, for example KH06 age: $(38 - 23) / (50 - 23) = 15 / 27 = 0.5556$; KH10 income: $(16 - 8) / (120 - 8) = 8 / 112 = 0.0714$; KH10 late payments: $3 / 4 = 0.75$.

- Lấy giá trị tuyệt đối của hiệu từng thuộc tính: 0,0741; 0,0179; 0,7500.

Take the absolute difference per attribute: 0.0741; 0.0179; 0.7500.

- Manhattan = 0,0741 + 0,0179 + 0,7500 = 0,8419.

Manhattan = 0.0741 + 0.0179 + 0.7500 = 0.8419.

- Euclid = $\sqrt{(0,0741^2 + 0,0179^2 + 0,7500^2)} = \sqrt{0,5683} = 0,7539$.

Euclid = $\sqrt{(0.0741^2 + 0.0179^2 + 0.7500^2)} = \sqrt{0.5683} = 0.7539$.

- Chebyshev = max(0,0741; 0,0179; 0,7500) = 0,75.

Chebyshev = max(0.0741; 0.0179; 0.7500) = 0.75.

PHẢI THUỘC / MEMORISE — Chuẩn hóa min–max bằng min, max của cả cột trước khi tính khoảng cách. Thứ tự: Chebyshev $\leq$ Minkowski bậc 3 $\leq$ Euclid $\leq$ Manhattan. Metric cần đủ bốn tính chất, trong đó có bất đẳng thức tam giác.

Apply min–max with the min and max of the whole column before computing distances. Order: Chebyshev $\leq$ Minkowski of order 3 $\leq$ Euclid $\leq$ Manhattan. A metric needs all four properties, including the triangle inequality.

LỖI HAY GẶP / COMMON MISTAKE — Quên chuẩn hóa nên thu nhập (8 đến 120) lấn át số lần trễ hạn (0 đến 4); lấy min, max của chỉ hai khách đang so sánh thay vì của cả cột; nhầm Manhattan với Euclid khi đề cho bậc h (hay p).

Forgetting to normalise, so income (8 to 120) swamps late payments (0 to 4); taking min and max of only the two customers being compared instead of the whole column; confusing Manhattan with Euclid when the problem gives the order h (or p).

2.7. Độ tương tự của thuộc tính nhị phân: SMC và Jaccard — 2.7. Similarity of binary attributes: SMC and Jaccard

Với p thuộc tính nhị phân, so sánh hai đối tượng i và j bằng **bảng tiếp liên 2×2** đếm số thuộc tính theo từng cặp giá trị: q là số thuộc tính cả hai cùng bằng 1; r là i bằng 1 còn j bằng 0; s là i bằng 0 còn j bằng 1; t là cả hai cùng bằng 0; $q + r + s + t = p$. Với KH02 và KH05 trên sáu cờ sản phẩm (thẻ tín dụng, vay tiền mặt, trả góp điện máy, trả góp xe máy, bảo hiểm khoản vay, dùng ứng dụng), KH02 dùng thẻ, vay tiền mặt, bảo hiểm và ứng dụng; KH05 dùng thẻ, xe máy và ứng dụng:

For p binary attributes, compare objects i and j with a 2×2 contingency table that counts the attributes by value pair: q is the number of attributes where both equal 1; r where i is 1 and j is 0; s where i is 0 and j is 1; t where both are 0; $q + r + s + t = p$. For KH02 and KH05 over six product flags (credit card, cash loan, electrical-appliance instalment, motorbike instalment, loan insurance, app use), KH02 uses the card, cash loan, insurance and app; KH05 uses the card, motorbike instalment and app:

Khách Customer	KH05 = 1	KH05 = 0	Cộng Total
KH02 = 1	$q = 2$ (thẻ, ứng dụng) $q = 2$ (card, app)	$r = 2$ (vay tiền mặt, bảo hiểm) $r = 2$ (cash loan, insurance)	4
KH02 = 0	$s = 1$ (xe máy) $s = 1$ (motorbike)	$t = 1$ (điện máy) $t = 1$ (appliance)	2
Cộng Total	3	3	6

$$SMC = (q + t) / (q + r + s + t); J = q / (q + r + s) \quad (2.14)$$

Hệ số khớp đơn giản SMC (simple matching coefficient) tính mọi thuộc tính khớp nhau, kể cả cặp 0-0. **Hệ số Jaccard** bỏ qua cặp 0-0. Khoảng cách tương ứng là $1 - SMC$ và $1 - J$. Chọn SMC cho thuộc tính nhị phân đối xứng (cả hai giá trị đều mang thông tin, như có nhà); chọn Jaccard cho nhị phân bất đối xứng (chỉ giá trị 1 mang thông tin, như cờ dùng sản phẩm).

*The **simple matching coefficient** (SMC) counts every attribute where the two agree, including 0-0 pairs. The **Jaccard coefficient** ignores 0-0 pairs. The matching distances are $1 - SMC$ and $1 - J$. Choose SMC for symmetric binary attributes (both values informative, like home ownership); choose Jaccard for asymmetric binary attributes (only 1 informative, like product-use flags).*

Tiêu chí Criterion	SMC	Jaccard
Cặp 0-0 0-0 pairs	Tính là khớp Counted as a match	Bỏ qua Ignored
Loại thuộc tính Attribute type	Nhị phân đối xứng Symmetric binary	Nhị phân bất đối xứng Asymmetric binary

Tiêu chí Criterion	SMC	Jaccard
Khi thêm nhiều thuộc tính cả hai cùng 0 <i>When many attributes are 0 for both</i>	Tăng về 1, che mất khác biệt thật <i>Rises towards 1, hiding real differences</i>	Không đổi <i>Unchanged</i>
KH02 và KH05 <i>KH02 and KH05</i>	$3 / 6 = 0,5$ $3 / 6 = 0.5$	$2 / 5 = 0,4$ $2 / 5 = 0.4$

```

a = [1, 1, 0, 0, 1, 1]      # KH02: thẻ, vay tiền mặt, bảo hiểm, ứng dụng
b = [1, 0, 0, 1, 0, 1]      # KH05: thẻ, xe máy, ứng dụng

def do_tuong_tu(a, b):
    q = sum(1 for x, y in zip(a, b) if x == 1 and y == 1)
    r = sum(1 for x, y in zip(a, b) if x == 1 and y == 0)
    s = sum(1 for x, y in zip(a, b) if x == 0 and y == 1)
    t = sum(1 for x, y in zip(a, b) if x == 0 and y == 0)
    return q, r, s, t, (q + t) / len(a), q / (q + r + s)

q, r, s, t, smc, jac = do_tuong_tu(a, b)
print("q, r, s, t =", q, r, s, t)
print("SMC =", round(smc, 4), "| Jaccard =", round(jac, 4))
a2, b2 = a + [0] * 10, b + [0] * 10      # thêm 10 sản phẩm cả hai không dùng
q, r, s, t, smc, jac = do_tuong_tu(a2, b2)
print("thêm 10 cặp 0-0: SMC =", round(smc, 4), "| Jaccard =", round(jac, 4))

```

Kết quả chạy / Output

```

q, r, s, t = 2 2 1 1
SMC = 0.5 | Jaccard = 0.4
thêm 10 cặp 0-0: SMC = 0.8125 | Jaccard = 0.4

```

Kết quả khớp bảng tiếp liên: $q = 2$, $r = 2$, $s = 1$, $t = 1$, $SMC = 0,5$ và $Jaccard = 0,4$. Khi giả sử có thêm 10 sản phẩm mà cả hai khách đều không dùng, SMC nhảy lên 0,8125 trong khi Jaccard vẫn 0,4: hai khách không giống nhau hơn chỉ vì cùng không dùng những sản phẩm họ chưa từng biết đến, nên Jaccard hợp với cờ dùng sản phẩm.

The result matches the contingency table: $q = 2$, $r = 2$, $s = 1$, $t = 1$, $SMC = 0.5$ and $Jaccard = 0.4$. When 10 more products that neither customer uses are assumed, SMC jumps to 0.8125 while Jaccard stays at 0.4: the customers are not more alike merely because both skip products they have never met, so Jaccard suits product-use flags.

Bài mẫu 2.6. SMC và Jaccard của KH02 và KH05 — *Worked example 2.6. SMC and Jaccard of KH02 and KH05*

Đề bài: Cho cờ sản phẩm KH02 = (1, 1, 0, 0, 1, 1) và KH05 = (1, 0, 0, 1, 0, 1). Tính SMC, Jaccard và hai khoảng cách tương ứng.

Problem: the product flags are KH02 = (1, 1, 0, 0, 1, 1) and KH05 = (1, 0, 0, 1, 0, 1). Compute SMC, Jaccard and the two matching distances.

- So từng vị trí: (1,1) ở vị trí 1 và 6 nên $q = 2$; (1,0) ở vị trí 2 và 5 nên $r = 2$; (0,1) ở vị trí 4 nên $s = 1$; (0,0) ở vị trí 3 nên $t = 1$. Kiểm: $2 + 2 + 1 + 1 = 6$.

Compare position by position: (1,1) at positions 1 and 6 gives $q = 2$; (1,0) at positions 2 and

5 gives $r = 2$; $(0,1)$ at position 4 gives $s = 1$; $(0,0)$ at position 3 gives $t = 1$. Check: $2 + 2 + 1 + 1 = 6$.

2. $SMC = (q + t) / p = (2 + 1) / 6 = 0,5$; khoảng cách $1 - SMC = 0,5$.

$SMC = (q + t) / p = (2 + 1) / 6 = 0.5$; distance $1 - SMC = 0.5$.

3. Jaccard $= q / (q + r + s) = 2 / (2 + 2 + 1) = 2 / 5 = 0,4$; khoảng cách $1 - J = 0,6$.

Jaccard $= q / (q + r + s) = 2 / (2 + 2 + 1) = 2 / 5 = 0.4$; distance $1 - J = 0.6$.

PHẢI THUỘC / MEMORISE — $SMC = (q + t) / p$ cho nhị phân đối xứng; Jaccard $= q / (q + r + s)$ cho nhị phân bất đối xứng. Hai hệ số chỉ trùng nhau khi $t = 0$.

$SMC = (q + t) / p$ for symmetric binary data; Jaccard $= q / (q + r + s)$ for asymmetric binary data. The two coincide only when $t = 0$.

LỖI HAY GẶP / COMMON MISTAKE — Dùng SMC cho cờ sản phẩm nên độ giống bị thổi phồng; để mẫu số Jaccard là p thay vì $q + r + s$; nhầm hệ số tương tự với khoảng cách (quên lấy 1 trừ).

Using SMC for product flags so similarity is inflated; putting p instead of $q + r + s$ in the Jaccard denominator; confusing a similarity coefficient with a distance (forgetting to subtract from 1).

2.8. Thuộc tính định danh, thứ bậc và độ tương tự cosine — 2.8. Nominal and ordinal attributes and cosine similarity

Thuộc tính định danh với nhiều giá trị (nghe) so bằng cách đếm số thuộc tính trùng khớp m trong p thuộc tính: $d(i, j) = (p - m) / p$. Cách khác là mã hóa one-hot (Chương 3) rồi dùng độ đo nhị phân. **Thuộc tính thứ bậc** (hạng thẻ) được đổi hạng về $[0, 1]$ theo công thức (2.15) rồi dùng khoảng cách của số.

A **nominal attribute** with many values (occupation) is compared by counting the matches m among p attributes: $d(i, j) = (p - m) / p$. Another way is one-hot encoding (Chapter 3) followed by a binary measure. An **ordinal attribute** (card tier) is mapped to $[0, 1]$ by formula (2.15) and then compared as a number.

$$d(i, j) = (p - m) / p; \quad z_{if} = (r_{if} - 1) / (M_f - 1) \quad (2.15)$$

Trong công thức (2.15), r là hạng của giá trị (1 đến M), M là số hạng của thuộc tính. Hạng thẻ có $M = 3$: Chuẩn có hạng 1 ($z = 0$), Vàng hạng 2 ($z = 0,5$), Bạch kim hạng 3 ($z = 1$).

In formula (2.15), r is the rank of the value (1 to M) and M is the number of ranks of the attribute. Card tier has $M = 3$: Standard has rank 1 ($z = 0$), Gold rank 2 ($z = 0.5$), Platinum rank 3 ($z = 1$).

```
hang = {"Chuẩn": 1, "Vàng": 2, "Bạch kim": 3}
M = len(hang)

def z(h):
    return (hang[h] - 1) / (M - 1)

def d_thu_bac(h1, h2):
    return abs(z(h1) - z(h2))

print("Chuẩn - Vàng:", d_thu_bac("Chuẩn", "Vàng"))
```

```
print("Chuẩn - Bạch kim:", d_thu_bac("Chuẩn", "Bạch kim"))
kh06 = ("Nhân viên văn phòng", "Vàng", 1)      # nghề, hạng thẻ, có nhà
kh10 = ("Công nhân", "Vàng", 1)
m = sum(1 for x, y in zip(kh06, kh10) if x == y)
p = len(kh06)
print("trùng khớp m =", m, "| d =", round((p - m) / p, 4))
```

Kết quả chạy / Output

```
Chuẩn - Vàng: 0.5
Chuẩn - Bạch kim: 1.0
trùng khớp m = 2 | d = 0.3333
```

Khoảng cách hạng thẻ giữa Chuẩn và Vàng là 0,5, giữa Chuẩn và Bạch kim là 1. Với ba thuộc tính định danh (nghề, hạng thẻ, có nhà) xét theo kiểu chỉ so bằng, KH06 và KH10 trùng hai thuộc tính (cùng Vàng, cùng có nhà) và khác nghề, nên $d = (3 - 2) / 3 = 0,3333$.

The card-tier distance between Standard and Gold is 0.5, between Standard and Platinum 1. Treating three attributes (occupation, card tier, home ownership) by equality only, KH06 and KH10 match on two (both Gold, both home owners) and differ in occupation, so $d = (3 - 2) / 3 = 0.3333$.

Độ tương tự cosine — Cosine similarity

Khi đối tượng là véc-tơ đếm (số giao dịch theo kênh, tần suất từ trong văn bản ở Chương 9), điều cần so là hướng của véc-tơ; độ dài không quan trọng. **Cosine** đo góc giữa hai véc-tơ; khoảng cách cosine là $1 - \text{cosine}$. Với véc-tơ không âm, cosine nằm trong $[0, 1]$.

*When objects are count vectors (transactions per channel, word frequencies in text in Chapter 9), what matters is the direction of the vectors; length is irrelevant. **Cosine** measures the angle between two vectors; the cosine distance is $1 - \text{cosine}$. For non-negative vectors the cosine lies in $[0, 1]$.*

$$\cos(x, y) = (x \cdot y) / (\|x\| * \|y\|); x \cdot y = \sum x_f * y_f; \|x\| = \sqrt{(\sum x_f^2)} \quad (2.16)$$

```
import math

a = [3, 0, 2, 1, 4]      # KH02: số giao dịch theo 5 kênh
b = [1, 2, 0, 3, 2]      # KH05
tich = sum(x * y for x, y in zip(a, b))
na = math.sqrt(sum(x * x for x in a))
nb = math.sqrt(sum(y * y for y in b))
print("a.b =", tich, "| |a| =", round(na, 4), "| |b| =", round(nb, 4))
cos = tich / (na * nb)
print("cosine =", round(cos, 4), "| khoảng cách =", round(1 - cos, 4))
gap = [2 * x for x in a]      # KH02 giao dịch gấp đôi mọi kênh
na2 = math.sqrt(sum(x * x for x in gap))
tich2 = sum(x * y for x, y in zip(gap, b))
print("cosine(2a, b) =", round(tich2 / (na2 * nb), 4))
```

Kết quả chạy / Output

```
a.b = 14 | |a| = 5.4772 | |b| = 4.2426
cosine = 0.6025 | khoảng cách = 0.3975
cosine(2a, b) = 0.6025
```

KH02 và KH05 có cosine 0,6025 trên năm kênh giao dịch, khoảng cách cosine 0,3975. Khi KH02 giao dịch gấp đôi trên mọi kênh, cosine không đổi: độ đo chỉ nhìn tỷ lệ giữa các kênh, nên khác với khoảng cách Euclid, vốn tăng khi véc-tơ dài ra.

KH02 and KH05 have a cosine of 0.6025 over five transaction channels, a cosine distance of 0.3975. When KH02 transacts twice as much on every channel the cosine is unchanged: the measure sees only the ratios between channels, unlike the Euclid distance, which grows as the vectors lengthen.

Bài mẫu 2.7. Cosine của hai véc-tơ giao dịch — *Worked example 2.7. Cosine of two transaction vectors*

Đề bài: Số giao dịch theo năm kênh của KH02 là (3, 0, 2, 1, 4) và của KH05 là (1, 2, 0, 3, 2). Tính cosine và khoảng cách cosine.

Problem: the transactions over five channels are (3, 0, 2, 1, 4) for KH02 and (1, 2, 0, 3, 2) for KH05. Compute the cosine and the cosine distance.

Kênh Channel	a (KH02)	b (KH05)	a * b	a ²	b ²
1	3	1	3	9	1
2	0	2	0	0	4
3	2	0	0	4	0
4	1	3	3	1	9
5	4	2	8	16	4
Cộng Sum			14	30	18

- Tích vô hướng $a \cdot b = 14$; $\|a\|^2 = 30$, $\|b\|^2 = 18$.
Dot product $a \cdot b = 14$; $\|a\|^2 = 30$, $\|b\|^2 = 18$.
- $\|a\| = \sqrt{30} = 5,4772$; $\|b\| = \sqrt{18} = 4,2426$.
 $\|a\| = \sqrt{30} = 5.4772$; $\|b\| = \sqrt{18} = 4.2426$.
- $\text{cosine} = 14 / (5,4772 * 4,2426) = 14 / 23,2379 = 0,6025$.
 $\text{cosine} = 14 / (5.4772 * 4.2426) = 14 / 23.2379 = 0.6025$.
- Khoảng cách cosine $= 1 - 0,6025 = 0,3975$.
 $\text{Cosine distance} = 1 - 0.6025 = 0.3975$.

LỖI HAY GẶP / COMMON MISTAKE — Dùng cosine khi độ lớn cũng quan trọng (khách giao dịch gấp đôi bị coi là giống hệt); tính $\|x\|$ mà quên căn bậc hai; báo cosine như khoảng cách.

Using cosine when magnitude also matters (a customer who transacts twice as much is treated as identical); computing $\|x\|$ without the square root; reporting the cosine as a distance.

2.9. Thuộc tính hỗn hợp và khoảng cách Gower — *2.9. Mixed attributes and the Gower distance*

Bảng khách hàng trộn số, nhị phân, định danh và thứ bậc trong cùng một dòng. **Khoảng cách Gower** giải bài toán này bằng cách tính khoảng cách thành phần dij(f) cho từng thuộc tính f, đều đưa về [0, 1], rồi lấy trung bình có trọng số. Trọng số dij(f) bằng 0 khi

một trong hai giá trị bị thiếu (hoặc cặp 0-0 của nhị phân bất đối xứng) và bằng 1 trong các trường hợp còn lại.

The customer table mixes numeric, binary, nominal and ordinal attributes in one row. The Gower distance handles this by computing a component distance $d_{ij}(f)$ for each attribute f , all scaled to $[0, 1]$, then taking a weighted average. The weight $\delta_{ij}(f)$ is 0 when either value is missing (or for a 0-0 pair of an asymmetric binary attribute) and 1 otherwise.

$$d(i, j) = \sum \delta_{ij}(f) * d_{ij}(f) / \sum \delta_{ij}(f) \quad (2.17)$$

Kiểu thuộc tính <i>Attribute type</i>	Khoảng cách thành phần <i>Component distance</i>	Ví dụ <i>Example</i>
Số <i>Numeric</i>	$ x_i - x_j / (\max - \min)$ <i>absolute difference / (max - min)</i>	Tuổi, thu nhập, dư nợ <i>Age, income, balance</i>
Định danh, nhị phân đối xứng <i>Nominal, symmetric binary</i>	0 nếu bằng nhau, 1 nếu khác <i>0 if equal, 1 if different</i>	Nghề, có nhà <i>Occupation, home ownership</i>
Nhị phân bất đối xứng <i>Asymmetric binary</i>	Như trên, bỏ qua cặp 0-0 ($\delta = 0$) <i>As above, 0-0 pairs ignored ($\delta = 0$)</i>	Cờ sản phẩm <i>Product flags</i>
Thứ bậc <i>Ordinal</i>	$ z_i - z_j $ với z đổi hạng về $[0, 1]$ <i>absolute difference of ranks z mapped to $[0, 1]$</i>	Hạng thẻ <i>Card tier</i>

```

lo = {"tuoi": 23, "thu_nhap": 8, "du_no": 10, "tre": 0}
hi = {"tuoi": 50, "thu_nhap": 120, "du_no": 150, "tre": 4}
bac = {"Chuẩn": 0, "Vàng": 0.5, "Bạch kim": 1} # (hạng - 1) / (M - 1)
kh04 = {"tuoi": 41, "thu_nhap": 22, "nghe": "Kinh doanh tự do",
        "hang": "Vàng", "du_no": 45, "co_nha": 1, "tre": 1}
kh07 = {"tuoi": 45, "thu_nhap": None, "nghe": "Kinh doanh tự do",
        "hang": "Vàng", "du_no": 40, "co_nha": 1, "tre": 2}
tong, so = 0.0, 0
for f in kh04:
    a, b = kh04[f], kh07[f]
    if a is None or b is None:
        print(f"{f:<9} bỏ qua (thiếu giá trị)")
        continue
    if f in lo:
        dif = abs(a - b) / (hi[f] - lo[f])
    elif f == "hang":
        dif = abs(bac[a] - bac[b])
    else:
        dif = 0 if a == b else 1
    tong += dif
    so += 1
    print(f"{f:<9} d = {dif:.4f}")
print("Gower =", round(tong / so, 4), "trên", so, "thuộc tính")

```

Kết quả chạy / Output

```

tuoi      d = 0.1481
thu_nhap  bỏ qua (thiếu giá trị)
nghe      d = 0.0000
hang      d = 0.0000
du_no     d = 0.0357
co_nha    d = 0.0000
tre       d = 0.2500
Gower = 0.0723 trên 6 thuộc tính

```

Ví dụ KH04 và KH07: thu nhập của KH07 thiếu nên bị bỏ qua; nghề, hạng thẻ và có nhà đều trùng (0); tuổi lệch $4 / 27 = 0,1481$; dư nợ lệch $5 / 140 = 0,0357$; số lần trễ hạn lệch $1 / 4 = 0,25$. Tổng 0,4339 chia 6 thuộc tính còn dùng được, $d = 0,0723$: hai khách gần nhau.

Example KH04 and KH07: the income of KH07 is missing and is skipped; occupation, card tier and home ownership all match (0); age differs by $4 / 27 = 0.1481$; balance by $5 / 140 = 0.0357$; late payments by $1 / 4 = 0.25$. The total 0.4339 divided by the 6 usable attributes gives $d = 0.0723$: the two customers are close.

Thư viện scikit-learn có sẵn hàm tính khoảng cách theo cặp trong mô-đun metrics, với các độ đo manhattan, euclidean, cosine, jaccard; thư viện SciPy có mô-đun spatial.distance với cityblock, chebyshev, minkowski. Khoảng cách Gower thường tự cài đặt hoặc dùng gói bổ sung. Nên tự tay tính một vài cặp như trên để biết kết quả của thư viện nghĩa là gì.

scikit-learn has a pairwise distance function in its metrics module, with the metrics manhattan, euclidean, cosine and jaccard; SciPy has the spatial.distance module with cityblock, chebyshev and minkowski. The Gower distance is usually written by hand or taken from an add-on package. Compute a few pairs by hand as above to know what a library result means.

Bài mẫu 2.8. Khoảng cách Gower giữa KH04 và KH07 — *Worked example 2.8. Gower distance between KH04 and KH07*

Đề bài: Tính khoảng cách Gower giữa KH04 và KH07 trên bảy thuộc tính của bảng khách hàng; min và max lấy trên cả 12 khách (tuổi 23 đến 50, dư nợ 10 đến 150, số lần trễ hạn 0 đến 4).

Problem: compute the Gower distance between KH04 and KH07 over the seven attributes of the customer table; min and max are taken over all 12 customers (age 23 to 50, balance 10 to 150, late payments 0 to 4).

Thuộc tính <i>Attribute</i>	KH04	KH07	δ	Khoảng cách thành phần <i>Component distance</i>
Tuổi <i>Age</i>	41	45	1	$4 / 27 = 0,1481$
Thu nhập <i>Income</i>	22	—	0	bỏ qua <i>skipped</i>
Nghề <i>Occupation</i>	Kinh doanh tự do <i>Self-employed</i>	Kinh doanh tự do <i>Self-employed</i>	1	0
Hạng thẻ <i>Card tier</i>	Vàng <i>Gold</i>	Vàng <i>Gold</i>	1	0
Dư nợ <i>Balance</i>	45	40	1	$5 / 140 = 0,0357$
Có nhà <i>Home</i>	1	1	1	0

Thuộc tính <i>Attribute</i>	KH04	KH07	δ	Khoảng cách thành phần <i>Component distance</i>
Trễ hạn <i>Late</i>	1	2	1	$1 / 4 = 0,25$

1. Với mỗi thuộc tính, xác định δ (0 nếu thiếu) và khoảng cách thành phần như bảng trên; hạng thẻ giống nhau nên z bằng nhau và hiệu bằng 0.

For each attribute decide δ (0 if missing) and the component distance as in the table; the card tiers are equal so the z values match and the difference is 0.

2. Tử số = $0,1481 + 0,0357 + 0,25 = 0,4339$; mẫu số = tổng $\delta = 6$.

Numerator = $0.1481 + 0.0357 + 0.25 = 0.4339$; denominator = sum of $\delta = 6$.

3. $d(KH04, KH07) = 0,4339 / 6 = 0,0723$.

$d(KH04, KH07) = 0.4339 / 6 = 0.0723$.

PHẢI THUỘC / MEMORISE — Gower = trung bình các khoảng cách thành phần trong $[0, 1]$, chỉ tính trên thuộc tính có giá trị ở cả hai đối tượng; mẫu số là tổng δ chứ không phải p .

Gower = average of component distances in $[0, 1]$, over the attributes that have a value in both objects; the denominator is the sum of δ , not p .

LỖI HAY GẶP / COMMON MISTAKE — Chia cho 7 thay vì 6 khi một thuộc tính bị thiếu; quên chia hiệu số cho $(\max - \min)$; coi hạng thẻ là định danh nên mất thông tin thứ bậc.

Dividing by 7 instead of 6 when an attribute is missing; forgetting to divide the difference by $(\max - \min)$; treating card tier as nominal and losing the ordering.

2.10. Bảng thuật ngữ — 2.10. Glossary

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Đối tượng	Object	Một dòng của bảng dữ liệu, ở đây là một khách hàng <i>One row of the data table, here one customer</i>
Thuộc tính	Attribute	Một cột của bảng dữ liệu, mô tả một đặc điểm của đối tượng <i>One column of the data table, describing a feature of the object</i>
Thuộc tính định danh	Nominal attribute	Giá trị là nhãn không có thứ tự, chỉ so sánh bằng hoặc khác <i>Values are labels with no order; only equal or not equal</i>
Nhị phân bất đối xứng	Asymmetric binary	Hai giá trị, chỉ giá trị 1 mang thông tin, đo bằng Jaccard <i>Two values, only 1 is informative, measured with Jaccard</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Thuộc tính thứ bậc	Ordinal attribute	Có thứ tự giữa các giá trị nhưng khoảng cách không xác định <i>Values are ordered but gaps are undefined</i>
Thuộc tính số khoảng	Interval attribute	Hiệu có nghĩa, không có điểm không tuyệt đối <i>Differences are meaningful, no true zero</i>
Thuộc tính số tỷ lệ	Ratio attribute	Có điểm không tuyệt đối, tỷ số có nghĩa <i>Has a true zero, ratios are meaningful</i>
Trung bình	Mean	Tổng các giá trị chia số quan sát <i>Sum of the values divided by the number of observations</i>
Trung vị	Median	Giá trị ở giữa dãy đã sắp xếp, không bị ngoại lai kéo lệch <i>Middle value of the sorted list, not pulled by outliers</i>
Mode	Mode	Giá trị xuất hiện nhiều nhất, có thể có nhiều mode <i>Most frequent value; there may be several</i>
Phương sai mẫu	Sample variance	Tổng bình phương lệch chia $n - 1$ <i>Sum of squared deviations divided by $n - 1$</i>
Độ lệch chuẩn	Standard deviation	Căn bậc hai của phương sai, cùng đơn vị với dữ liệu <i>Square root of the variance, in the unit of the data</i>
Tứ phân vị	Quartile	Q1, Q2, Q3 chia dãy đã sắp xếp thành bốn phần <i>Q1, Q2, Q3 split the sorted list into four parts</i>
IQR	Interquartile range	Hiệu $Q3 - Q1$, độ rộng của nửa giữa dữ liệu <i>The difference $Q3 - Q1$, the width of the middle half</i>
Ngoại lai	Outlier	Giá trị nằm ngoài $[Q1 - 1,5 * IQR; Q3 + 1,5 * IQR]$ <i>A value outside $[Q1 - 1.5 * IQR; Q3 + 1.5 * IQR]$</i>
Hình hộp	Boxplot	Biểu đồ năm số tóm tắt kèm các điểm ngoại lai <i>Chart of the five-number summary with outlier points</i>
Độ lệch	Skewness	Mức bất đối xứng của phân phối; dương là lệch phải <i>Asymmetry of a distribution; positive means right-skewed</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Hệ số Pearson	Pearson correlation coefficient	Đo quan hệ tuyến tính giữa hai thuộc tính số, từ -1 đến 1 <i>Measures the linear relation between two numeric attributes, from -1 to 1</i>
Ma trận khoảng cách	Dissimilarity matrix	Ma trận vuông đối xứng $n \times n$, đường chéo bằng 0 <i>Symmetric $n \times n$ square matrix with a zero diagonal</i>
Khoảng cách Minkowski	Minkowski distance	Họ khoảng cách bậc p : Manhattan (1), Euclid (2), Chebyshev (∞) <i>Family of order-p distances: Manhattan (1), Euclid (2), Chebyshev (∞)</i>
Hệ số khớp đơn giản	Simple matching coefficient (SMC)	$(q + t) / (q + r + s + t)$, cho nhị phân đối xứng <i>$(q + t) / (q + r + s + t)$, for symmetric binary data</i>
Hệ số Jaccard	Jaccard coefficient	$q / (q + r + s)$, bỏ qua cặp $0-0$, cho nhị phân bất đối xứng <i>$q / (q + r + s)$, ignores $0-0$ pairs, for asymmetric binary data</i>
Độ tương tự cosine	Cosine similarity	Cosin của góc giữa hai véc-tơ, bỏ qua độ dài <i>Cosine of the angle between two vectors, ignoring length</i>
Khoảng cách Gower	Gower distance	Trung bình các khoảng cách thành phần đã đưa về $[0, 1]$, dùng cho dữ liệu hỗn hợp <i>Average of the component distances scaled to $[0, 1]$, for mixed data</i>

2.11. Bài tập vận dụng có lời giải — *Applied exercises with solutions*

Bài 1. Thống kê mô tả khoản trả góp xe máy — *Exercise 1. Descriptive statistics of motorbike instalment loans*

Công ty Phú Sa giải ngân 9 khoản trả góp xe máy với số tiền (triệu đồng): 18, 22, 25, 25, 28, 30, 35, 40, 85 (dữ kiện đề bài). Tính trung bình, trung vị, mode, Q1, Q3, IQR, hàng rào ngoại lai và độ lệch chuẩn mẫu; cho biết khoản nào là ngoại lai và trung bình thay đổi thế nào khi bỏ khoản đó.

Phú Sa disbursed 9 motorbike instalment loans of the following amounts (VND million): 18, 22, 25, 25, 28, 30, 35, 40, 85 (problem data). Compute the mean, median, mode, Q1, Q3, IQR, outlier fences and sample standard deviation; state which loan is an outlier and how the mean changes when it is dropped.

1. Dãy đã sắp xếp sẵn, $n = 9$, tổng = 308. Trung bình = $308 / 9 = 34,22$ triệu đồng.

The list is already sorted, $n = 9$, sum = 308. Mean = $308 / 9 = 34.22$ million.

2. Trung vị: $n = 9$ lẻ nên là giá trị thứ 5, bằng 28. Mode là 25 (xuất hiện 2 lần, các giá trị khác 1 lần).
Median: $n = 9$ is odd, so it is the 5th value, 28. The mode is 25 (twice; every other value once).
3. Q1: vị trí $= 1 + 8 * 0,25 = 3$, giá trị thứ 3 là 25. Q3: vị trí $= 1 + 8 * 0,75 = 7$, giá trị thứ 7 là 35. $IQR = 35 - 25 = 10$.
Q1: position $= 1 + 8 * 0.25 = 3$, the 3rd value is 25. Q3: position $= 1 + 8 * 0.75 = 7$, the 7th value is 35. $IQR = 35 - 25 = 10$.
4. Hàng rào: dưới $= 25 - 1,5 * 10 = 10$; trên $= 35 + 15 = 50$. Khoản 85 vượt 50 nên là ngoại lai; khoản 18 không thấp hơn 10 nên không phải.
Fences: lower $= 25 - 1.5 * 10 = 10$; upper $= 35 + 15 = 50$. The loan of 85 exceeds 50 and is an outlier; 18 is not less than 10 and is not.
5. $\sum x^2 = 13.792$, $(\sum x)^2 / n = 10.540,44$, tổng bình phương lệch $= 3.251,56$; $s^2 = 3.251,56 / 8 = 406,44$; $s = 20,16$ triệu đồng.
 $\sum x^2 = 13,792$, $(\sum x)^2 / n = 10,540.44$, sum of squared deviations $= 3,251.56$; $s^2 = 3,251.56 / 8 = 406.44$; $s = 20.16$ million.
6. Bỏ khoản 85: trung bình $= (308 - 85) / 8 = 27,88$, giảm từ 34,22; trung vị chuyển từ 28 sang 26,5 (trung bình của giá trị thứ 4 và 5 là 25 và 28).
Dropping the loan of 85: mean $= (308 - 85) / 8 = 27.88$, down from 34.22; the median moves from 28 to 26.5 (average of the 4th and 5th values, 25 and 28).

Bài 2. Chuẩn hóa và so khoảng cách của ba khách — Exercise 2. Normalisation and distance comparison for three customers

Ba khách KH21 (30 tuổi, thu nhập 16 triệu đồng), KH22 (45 tuổi, 26 triệu đồng) và KH23 (30 tuổi, 36 triệu đồng) (dữ kiện đề bài). Toàn bộ khách có tuổi từ 20 đến 60 và thu nhập từ 6 đến 46 triệu đồng. Chuẩn hóa min–max rồi tìm khách gần KH21 nhất theo Euclid, Manhattan và Chebyshev.

Three customers: KH21 (age 30, income VND 16 million), KH22 (age 45, VND 26 million) and KH23 (age 30, VND 36 million) (problem data). All customers have ages from 20 to 60 and incomes from 6 to 46 million. Normalise with min–max, then find the customer closest to KH21 under the Euclid, Manhattan and Chebyshev distances.

1. Chuẩn hóa: tuổi $(x - 20) / 40$, thu nhập $(x - 6) / 40$. KH21 = (0,25; 0,25); KH22 = (0,625; 0,5); KH23 = (0,25; 0,75).
Normalise: age $(x - 20) / 40$, income $(x - 6) / 40$. KH21 = (0.25, 0.25); KH22 = (0.625, 0.5); KH23 = (0.25, 0.75).
2. Chênh lệch tuyệt đối với KH21: KH22 có (0,375; 0,25); KH23 có (0; 0,5).
Absolute differences from KH21: KH22 has (0.375, 0.25); KH23 has (0, 0.5).
3. Euclid: $d(KH21, KH22) = \sqrt{(0,375^2 + 0,25^2)} = 0,4507$; $d(KH21, KH23) = 0,50$. Gần nhất là KH22.
Euclid: $d(KH21, KH22) = \sqrt{(0.375^2 + 0.25^2)} = 0.4507$; $d(KH21, KH23) = 0.50$. The closest is KH22.

4. Manhattan: $d(KH21, KH22) = 0,375 + 0,25 = 0,625$; $d(KH21, KH23) = 0 + 0,5 = 0,5$. Gần nhất là KH23.
Manhattan: $d(KH21, KH22) = 0.375 + 0.25 = 0.625$; $d(KH21, KH23) = 0 + 0.5 = 0.5$. The closest is KH23.
5. Chebyshev: $d(KH21, KH22) = \max(0,375; 0,25) = 0,375$; $d(KH21, KH23) = \max(0; 0,5) = 0,5$. Gần nhất là KH22.
Chebyshev: $d(KH21, KH22) = \max(0.375, 0.25) = 0.375$; $d(KH21, KH23) = \max(0, 0.5) = 0.5$. The closest is KH22.
6. Nhận xét: khách gần nhất đổi theo độ đo (KH22 với Euclid và Chebyshev, KH23 với Manhattan), nên phải nêu độ đo khi kết luận về láng giềng.
Comment: the nearest customer depends on the measure (KH22 under Euclid and Chebyshev, KH23 under Manhattan), so the measure must be stated when concluding about neighbours.

Bài 3. Độ tương tự nhị phân và cosine của hai khách — Exercise 3. Binary similarity and cosine of two customers

Hai khách KH31 và KH32 có tám cờ dịch vụ (1 là đang dùng): KH31 = (1, 0, 0, 1, 1, 0, 0, 0), KH32 = (1, 1, 0, 0, 1, 0, 0, 0); số giao dịch trên bốn kênh là KH31 = (2, 0, 1, 3), KH32 = (1, 1, 0, 2) (dữ kiện đề bài). Lập bảng tiếp liên, tính SMC, Jaccard và cosine; cho biết nên dùng hệ số nào cho cờ dịch vụ.

Two customers KH31 and KH32 have eight service flags (1 means in use): KH31 = (1, 0, 0, 1, 1, 0, 0, 0), KH32 = (1, 1, 0, 0, 1, 0, 0, 0); their transactions over four channels are KH31 = (2, 0, 1, 3), KH32 = (1, 1, 0, 2) (problem data). Build the contingency table, compute SMC, Jaccard and cosine, and say which coefficient to use for service flags.

1. Đếm từng vị trí: cả hai bằng 1 ở vị trí 1 và 5 nên $q = 2$; KH31 bằng 1 còn KH32 bằng 0 ở vị trí 4 nên $r = 1$; KH31 bằng 0 còn KH32 bằng 1 ở vị trí 2 nên $s = 1$; cả hai bằng 0 ở bốn vị trí còn lại nên $t = 4$. Kiểm: $2 + 1 + 1 + 4 = 8$.
Count by position: both are 1 at positions 1 and 5, so $q = 2$; KH31 is 1 and KH32 is 0 at position 4, so $r = 1$; KH31 is 0 and KH32 is 1 at position 2, so $s = 1$; both are 0 at the other four positions, so $t = 4$. Check: $2 + 1 + 1 + 4 = 8$.
2. $SMC = (2 + 4) / 8 = 0,75$.
 $SMC = (2 + 4) / 8 = 0.75$.
3. $Jaccard = 2 / (2 + 1 + 1) = 0,5$.
 $Jaccard = 2 / (2 + 1 + 1) = 0.5$.
4. Cosine: tích vô hướng = 8; $\|a\| = \sqrt{14}$; $\|b\| = \sqrt{6}$; $cosine = 8 / \sqrt{84} = 8 / 9,1652 = 0,8729$.
Cosine: dot product = 8; $\|a\| = \sqrt{14}$; $\|b\| = \sqrt{6}$; $cosine = 8 / \sqrt{84} = 8 / 9.1652 = 0.8729$.
5. Cờ dịch vụ là nhị phân bất đối xứng (chỉ giá trị 1 mang thông tin) nên dùng Jaccard (0,5); SMC bị đẩy lên 0,75 bởi bốn cặp 0-0.
Service flags are asymmetric binary (only 1 is informative), so use Jaccard (0.5); SMC is pushed up to 0.75 by the four 0-0 pairs.

2.12. Câu hỏi tự luận và dàn ý trả lời — *Essay questions and answer outlines*

Câu 1. Phân biệt năm kiểu thuộc tính (định danh, nhị phân, thứ bậc, số khoảng, số tỷ lệ) và nêu phép toán hợp lệ của từng kiểu. Lấy ví dụ từ bảng khách hàng Phú Sa.

Question 1. *Distinguish the five attribute types (nominal, binary, ordinal, interval, ratio) and state the valid operations of each. Use examples from the Phú Sa customer table.*

- Định danh (nghề): chỉ so bằng hoặc khác, đếm, mode; không có thứ tự.
Nominal (occupation): only equal or not equal, counting, mode; no order.
- Nhị phân: đối xứng (có nhà) và bất đối xứng (cờ dùng sản phẩm), kéo theo SMC hay Jaccard.
Binary: symmetric (home ownership) and asymmetric (product-use flags), leading to SMC or Jaccard.
- Thứ bậc (hạng thẻ): có thứ tự, thêm trung vị và tứ phân vị, nhưng khoảng cách giữa các hạng không xác định.
Ordinal (card tier): ordered, adds the median and quartiles, but the gaps between tiers are undefined.
- Số khoảng (năm ký hợp đồng): hiệu có nghĩa, thêm trung bình và độ lệch chuẩn, tỷ số vô nghĩa vì không có điểm không tuyệt đối.
Interval (contract year): differences are meaningful, adds the mean and standard deviation, ratios are meaningless without a true zero.
- Số tỷ lệ (thu nhập, dư nợ): có điểm không tuyệt đối, mọi phép toán kể cả tỷ số; ví dụ thu nhập 18 gấp đôi 9.
Ratio (income, balance): true zero, all operations including ratios; for example income 18 is twice 9.
- Lỗi hay gặp: mã hóa hạng thẻ thành 1, 2, 3 rồi tính trung bình như số tỷ lệ.
Common mistake: coding card tier as 1, 2, 3 and averaging it as a ratio attribute.

Câu 2. Vì sao khi dữ liệu có ngoại lai nên báo trung vị và IQR cùng với (hoặc thay cho) trung bình và độ lệch chuẩn? Minh họa bằng thu nhập của 11 khách Phú Sa.

Question 2. *Why should the median and IQR be reported together with (or instead of) the mean and standard deviation when the data contain outliers? Illustrate with the income of 11 Phú Sa customers.*

- Trung bình và độ lệch chuẩn dùng mọi giá trị nên một ngoại lai kéo chúng lệch; trung vị và IQR chỉ dựa trên thứ hạng.
The mean and standard deviation use every value, so one outlier drags them; the median and IQR rest on ranks only.
- Số liệu: trung bình 24,09, trung vị 15; độ lệch chuẩn 32,11 so với IQR 7,5; bỏ KH12 thì trung bình còn 14,5.
Figures: mean 24.09, median 15; standard deviation 32.11 against IQR 7.5; without KH12 the mean falls to 14.5.
- Hình hộp và quy tắc $1,5 * IQR$ chỉ ra KH12 là ngoại lai (hàng rào 30,25); histogram cho thấy phân phối lệch phải.

The boxplot and the $1.5 * IQR$ rule flag KH12 as an outlier (fence 30.25); the histogram shows a right-skewed distribution.

- Không tự động xóa ngoại lai: KH12 có thể là khách thật có thu nhập cao hoặc lỗi nhập; cần đối chiếu nguồn rồi mới quyết định.

Do not delete outliers automatically: KH12 may be a genuine high earner or a data-entry error; check the source before deciding.

- Khi báo cáo, nêu cả hai nhóm số và quy ước tứ phân vị đã dùng.

In a report, give both groups of figures and the quartile convention used.

Câu 3. Công ty Phú Sa cần đo độ giống nhau giữa hai khách có dữ liệu hỗn hợp (số, nhị phân, định danh, thứ bậc, có ô thiếu). Trình bày cách chọn và tính độ đo.

Question 3. *Phú Sa needs to measure how alike two customers are when the data are mixed (numeric, binary, nominal, ordinal, with missing cells). Describe how to choose and compute the measure.*

- Với riêng thuộc tính số: chuẩn hóa min–max (min, max của cả cột) rồi dùng Minkowski; nêu rõ bậc đã chọn.

For numeric attributes alone: min–max normalisation (min and max of the whole column), then Minkowski; state the order chosen.

- Với riêng cờ nhị phân bất đối xứng dùng Jaccard, nhị phân đối xứng dùng SMC; với véc-tơ đếm mà hướng quan trọng hơn độ lớn thì dùng cosine.

For asymmetric binary flags alone use Jaccard, for symmetric binary use SMC; for count vectors where direction matters more than magnitude use cosine.

- Với thuộc tính hỗn hợp dùng khoảng cách Gower: mỗi thuộc tính cho một khoảng cách thành phần trong $[0, 1]$.

For mixed attributes use the Gower distance: each attribute gives a component distance in $[0, 1]$.

- Thuộc tính thiếu giá trị có trọng số δ bằng 0 và bị loại khỏi cả tử số lẫn mẫu số.

A missing attribute has weight δ equal to 0 and is dropped from both numerator and denominator.

- Thuộc tính thứ bậc đổi hạng về $[0, 1]$ bằng $(r - 1) / (M - 1)$ thay vì coi là định danh.

An ordinal attribute is mapped to $[0, 1]$ by $(r - 1) / (M - 1)$ instead of being treated as nominal.

- Kiểm lại: khoảng cách phải thỏa bất đẳng thức tam giác nếu thuật toán phía sau đòi hỏi metric.

Check: the distance should satisfy the triangle inequality if the downstream algorithm requires a metric.

2.13. Câu hỏi trắc nghiệm — Multiple-choice questions

9. Thuộc tính hạng thẻ của Phú Sa (Chuẩn, Vàng, Bạch kim) thuộc kiểu nào?
Which type is the card tier attribute of Phú Sa (Standard, Gold, Platinum)?
- A. Định danh / *Nominal*
 - B. Thứ bậc / *Ordinal*
 - C. Số khoảng / *Interval*
 - D. Số tỷ lệ / *Ratio*
10. Dư nợ của 6 khoản (triệu đồng): 12, 14, 18, 22, 30, 90. Trung vị là bao nhiêu?
The balances of 6 loans (VND million) are 12, 14, 18, 22, 30, 90. What is the median?
- A. 18
 - B. 19
 - C. 20
 - D. 31
11. Phương sai mẫu của ba giá trị 2, 4, 6 là bao nhiêu?
What is the sample variance of the three values 2, 4, 6?
- A. 2,67 / *2.67*
 - B. 8
 - C. 4
 - D. 2
12. Biết $Q1 = 12$ và $Q3 = 20$. Hàng rào ngoại lai trên theo quy tắc $1,5 * IQR$ là bao nhiêu?
Given $Q1 = 12$ and $Q3 = 20$, what is the upper outlier fence under the $1.5 * IQR$ rule?
- A. 28
 - B. 32
 - C. 36
 - D. 44
13. Hệ số Jaccard phù hợp nhất với loại thuộc tính nào?
Which attribute type is the Jaccard coefficient best suited to?
- A. Định danh nhiều giá trị như nghề / *A nominal attribute with many values, such as occupation*
 - B. Số tỷ lệ đã chuẩn hóa về đoạn từ 0 đến 1 / *A ratio attribute normalised to the range 0 to 1*
 - C. Nhị phân đối xứng như có nhà hay không / *A symmetric binary attribute such as owning a home*
 - D. Nhị phân bất đối xứng như cờ dùng sản phẩm / *An asymmetric binary attribute such as a product-use flag*
14. Sau chuẩn hóa, hai khách là (0,2; 0,5; 0,9) và (0,6; 0,4; 0,1). Khoảng cách Manhattan giữa họ là bao nhiêu?

After normalisation two customers are (0.2, 0.5, 0.9) and (0.6, 0.4, 0.1). What is the Manhattan distance between them?

- A.** 0,8 / 0.8
- B.** 0,9 / 0.9
- C.** 1,3 / 1.3
- D.** 2,1 / 2.1

- 15.** Với st là module statistics của Python, câu lệnh "print(round(st.pstdev([2, 4, 6]), 2))" in ra giá trị nào?

With st as the Python statistics module, which value does "print(round(st.pstdev([2, 4, 6]), 2))" print?

- A.** 2.0
- B.** 1.63
- C.** 2.45
- D.** 4.0

- 16.** Thu nhập có trung bình 24,09 và trung vị 15. Phân phối của thu nhập này như thế nào?

Income has a mean of 24.09 and a median of 15. What is the shape of this distribution?

- A.** Đối xứng quanh trung bình / *Symmetric around the mean*
- B.** Lệch phải / *Right-skewed*
- C.** Lệch trái / *Left-skewed*
- D.** Hai đỉnh cân bằng / *Bimodal and balanced*

Chương 3. Tiền xử lý dữ liệu — Chapter 3. Data preprocessing

Mục tiêu chương — Chapter objectives

- Đánh giá chất lượng dữ liệu theo sáu tiêu chí và xếp các thao tác tiền xử lý vào bốn nhóm: làm sạch, tích hợp, biến đổi, thu giảm.
Assess data quality on six dimensions and place preprocessing operations into four groups: cleaning, integration, transformation, reduction.
- Xử lý giá trị thiếu (11 khách có thu nhập, KH07 thiếu), làm trơn nhiễu bằng giỏ đều độ rộng và đều tần số, phát hiện ngoại lai bằng IQR và z-score.
Handle missing values (11 customers with income, KH07 missing), smooth noise with equal-width and equal-frequency bins, and detect outliers by IQR and z-score.
- Tích hợp dữ liệu: nhận diện trùng thực thể, xung đột đơn vị, thuộc tính dư thừa bằng tương quan Pearson và kiểm định χ^2 trên bảng 2×2 .
Integrate data: recognise entity duplicates, unit conflicts and redundant attributes through Pearson correlation and the χ^2 test on a 2×2 table.
- Biến đổi và thu giảm: chuẩn hóa min–max, z-score, thập phân; one-hot và mã hóa thứ bậc; DTI; chọn thuộc tính; PCA hai chiều tính bằng công thức đóng.
Transform and reduce: min–max, z-score and decimal scaling; one-hot and ordinal encoding; DTI; feature selection; two-dimensional PCA by closed formula.
- Lấy mẫu phân tầng với random seed 2026, xử lý mất cân bằng lớp (giảm mẫu, tăng mẫu, SMOTE) và tránh rò rỉ dữ liệu khi chuẩn hóa.
Draw stratified samples with random seed 2026, treat class imbalance (undersampling, oversampling, SMOTE) and avoid data leakage when normalising.

3.1. Chất lượng dữ liệu và các bước tiền xử lý — 3.1. Data quality and the preprocessing steps

Mô hình khai phá dữ liệu học từ những gì bảng dữ liệu chứa. Bảng sai thì mô hình sai theo, dù thuật toán viết đúng. Công ty Tài chính TNHH Phú Sa gom dữ liệu từ ba nơi: hệ thống cho vay, ứng dụng "Phú Sa" và hệ thống bán bảo hiểm khoản vay. Mỗi nơi ghi đơn vị, định dạng ngày và cách bỏ trống ô theo kiểu riêng, nên bảng gộp hiếm khi dùng được ngay. Tiền xử lý dữ liệu (data preprocessing) là nhóm thao tác đưa bảng gộp về dạng thuật toán chạy được và cho kết quả đáng tin.

A data mining model learns from what the table contains. A wrong table gives a wrong model even when the algorithm is correct. Phu Sa Finance Co., Ltd. gathers data from three places: the lending system, the "Phú Sa" app and the loan-insurance sales system. Each records units, date formats and blank cells in its own way, so the merged table is seldom usable as it stands. Data preprocessing is the set of operations that bring the merged table into a form the algorithms can run on and give results worth trusting.

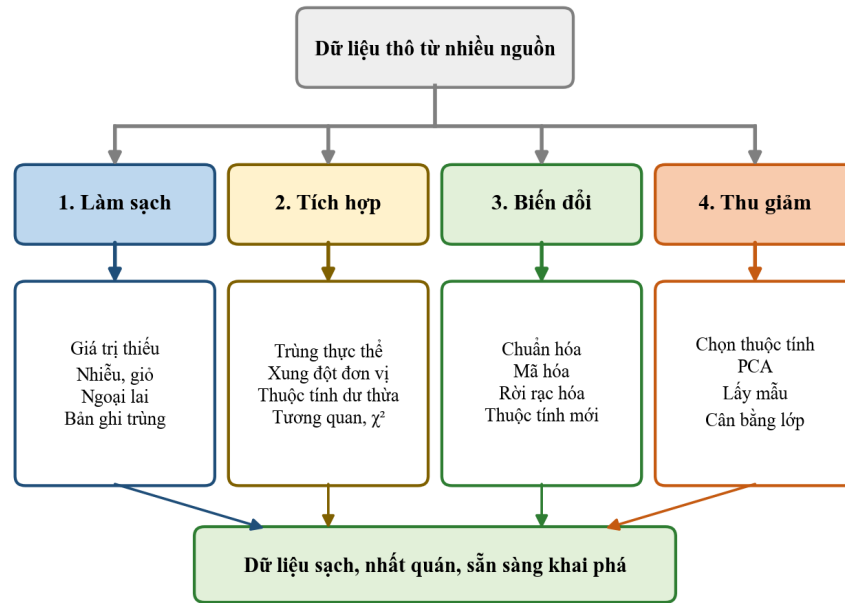
Chất lượng dữ liệu được đánh giá theo sáu tiêu chí. Bảng sau gắn mỗi tiêu chí với câu hỏi kiểm tra và một tình huống ở Phú Sa.

Data quality is judged on six dimensions. The table attaches to each one a check question and a situation at Phú Sa.

Tiêu chí <i>Dimension</i>	Câu hỏi kiểm tra <i>Check question</i>	Tình huống ở Phú Sa <i>Situation at Phú Sa</i>
Chính xác <i>Accuracy</i>	Giá trị có đúng với thực tế không? <i>Is the value true to reality?</i>	Tuổi ghi 245 do gõ thừa chữ số <i>Age recorded as 245 because of an extra typed digit</i>
Đầy đủ <i>Completeness</i>	Ô nào bị bỏ trống? <i>Which cells are blank?</i>	KH07 không có thu nhập <i>KH07 has no income</i>
Nhất quán <i>Consistency</i>	Cùng một thông tin ở hai nơi có khớp không? <i>Does the same fact agree in two places?</i>	Hạng thẻ "Vàng" ở hệ thống thẻ, "Gold" ở ứng dụng <i>Card tier "Vàng" in the card system, "Gold" in the app</i>
Kịp thời <i>Timeliness</i>	Dữ liệu có phản ánh tình trạng hiện tại không? <i>Does the data reflect the current state?</i>	Dư nợ chốt cuối tháng trước, khách đã trả thêm trong tháng <i>Balance fixed at last month's end, though the customer has paid more since</i>
Tin cậy <i>Believability</i>	Người dùng có tin nguồn dữ liệu không? <i>Do users trust the source?</i>	Thu nhập tự khai khác thu nhập trong bảng lương <i>Self-declared income differs from payroll income</i>
Diễn giải được <i>Interpretability</i>	Người đọc có hiểu mỗi cột nghĩa gì không? <i>Can a reader tell what each column means?</i>	Cột số lần trễ hạn không ghi kỳ đếm là 12 hay 24 tháng <i>The late-payment count does not say whether it covers 12 or 24 months</i>

Bốn nhóm công việc xử lý các lỗi đó theo thứ tự thường gặp trong dự án: làm sạch, tích hợp, biến đổi, thu giảm (Hình 3.1). Thứ tự không cứng. Muốn chia giỏ hoặc chuẩn hóa thì phải xử lý giá trị thiếu trước; muốn thu giảm số chiều thì phải biến đổi xong.

Four groups of work deal with those faults in the order usually met in a project: cleaning, integration, transformation and reduction (Figure 3.1). The order is not rigid. Binning or normalising needs the missing values dealt with first; reducing dimensions needs the transformation finished.



Hình 3.1. Bốn nhóm công việc tiền xử lý dữ liệu

Figure 3.1. The four groups of preprocessing work

Đoạn mã sau kiểm chất lượng bảng 12 khách hàng ở bốn khía cạnh: ô thiếu, mã trùng, tuổi ngoài miền hợp lệ và thu nhập bất thường.

The code audits the table of 12 customers on four points: missing cells, duplicate codes, ages outside the valid range and unusual income.

```

khach = [
    ("KH01", 25, 9, 12), ("KH02", 29, 12, 18), ("KH03", 34, 15, 20),
    ("KH04", 41, 22, 45), ("KH05", 23, 8, 10), ("KH06", 38, 18, 30),
    ("KH07", 45, None, 40), ("KH08", 31, 14, 16), ("KH09", 27, 11, 14),
    ("KH10", 36, 16, 25), ("KH11", 50, 20, 35), ("KH12", 47, 120, 150),
]
cot = ["ma", "tuoi", "thu_nhap", "du_no"]
thieu = {c: 0 for c in cot}
for dong in khach:
    for c, v in zip(cot, dong):
        if v is None:
            thieu[c] += 1
print("Số ô thiếu theo cột:", thieu)
o = len(khach) * len(cot)
ti_le = f"{sum(thieu.values()) / o * 100:.2f}%".replace(".", ",")
print("Tỷ lệ ô thiếu:", ti_le, "của", o, "ô")
print("Mã trùng:", len(khach) - len({d[0] for d in khach}))
print("Tuổi ngoài 18-70:", [d[0] for d in khach if not 18 <= d[1] <= 70])
print("Thu nhập trên 100 triệu, cần soát:",
      [d[0] for d in khach if d[2] and d[2] > 100])
    
```

Kết quả chạy / Output

```

Số ô thiếu theo cột: {'ma': 0, 'tuoi': 0, 'thu_nhap': 1, 'du_no': 0}
Tỷ lệ ô thiếu: 2,08% của 48 ô
Mã trùng: 0
Tuổi ngoài 18-70: []
Thu nhập trên 100 triệu, cần soát: ['KH12']
    
```

Kết quả: 1 trong 48 ô bị thiếu (2,08%), nằm ở cột thu nhập của KH07; không có mã trùng; không có tuổi ngoài khoảng 18 đến 70; KH12 có thu nhập 120 triệu đồng, vượt

ngưỡng 100 triệu đặt cho bước soát. "Cần soát" khác với "sai": KH12 có thể là chủ doanh nghiệp thật, và mục 3.4 cho thấy cách xử lý.

Result: 1 of 48 cells is missing (2.08%), in the income column of KH07; there are no duplicate codes and no ages outside 18 to 70; KH12 has an income of 120 million dong, above the 100 million threshold set for review. "Needs review" is not the same as "wrong": KH12 may be a genuine business owner, and section 3.4 shows how to handle that.

PHẢI THUỘC / MEMORISE — Sáu tiêu chí chất lượng: chính xác, đầy đủ, nhất quán, kịp thời, tin cậy, diễn giải được. Bốn nhóm tiền xử lý: làm sạch, tích hợp, biến đổi, thu giảm.
Six quality dimensions: accuracy, completeness, consistency, timeliness, believability, interpretability. Four preprocessing groups: cleaning, integration, transformation, reduction.

3.2. Làm sạch: giá trị thiếu — 3.2. Cleaning: missing values

Giá trị thiếu (missing value) xuất hiện vì khách không khai, nhân viên quên nhập, hai hệ thống không khớp khóa hoặc ô không áp dụng. Nguyên nhân quyết định cách xử lý. Thiếu thuần ngẫu nhiên (MCAR) thì ô trống không liên quan đến giá trị nào. Thiếu ngẫu nhiên có điều kiện (MAR) thì khả năng thiếu phụ thuộc cột khác, chẳng hạn khách kinh doanh tự do hay bỏ trống thu nhập. Thiếu không ngẫu nhiên (MNAR) thì chính giá trị làm ô bị bỏ trống, chẳng hạn khách thu nhập thấp ngại khai. Với MNAR, điền trung bình làm lệch kết quả; nên giữ thêm một cột cờ "có thiếu" làm thuộc tính.

A missing value arises when the customer does not declare, staff forget to enter, two systems fail to match on a key, or the cell does not apply. The cause decides the treatment. Missing completely at random (MCAR) means the blank has nothing to do with any value. Missing at random (MAR) means the chance of a blank depends on another column, for instance self-employed customers often leave income empty. Missing not at random (MNAR) means the value itself causes the blank, for instance low earners reluctant to declare. With MNAR, mean imputation distorts the result; keep an extra "was missing" flag column as an attribute.

Cách xử lý <i>Treatment</i>	Làm gì <i>What is done</i>	Ưu điểm <i>Strength</i>	Hạn chế <i>Weakness</i>
Bỏ dòng <i>Delete the row</i>	Loại khách có ô thiếu <i>Drop customers with a blank</i>	Đơn giản <i>Simple</i>	Mất mẫu; lệch nếu thiếu không ngẫu nhiên <i>Loses samples; biased if the gaps are not random</i>
Hằng số <i>Constant</i>	Điền một mã riêng, ví dụ 0 hoặc "chưa rõ" <i>Fill a dedicated code such as 0 or "unknown"</i>	Giữ cờ thiếu <i>Keeps the flag</i>	Số 0 bị hiểu là thu nhập thật <i>A 0 is read as real income</i>
Trung bình <i>Mean</i>	Điền trung bình cột <i>Fill the column mean</i>	Không đổi trung bình <i>Leaves the mean unchanged</i>	Nhạy ngoại lai; thu hẹp độ phân tán <i>Sensitive to outliers; shrinks the spread</i>

Cách xử lý <i>Treatment</i>	Làm gì <i>What is done</i>	Ưu điểm <i>Strength</i>	Hạn chế <i>Weakness</i>
Trung vị <i>Median</i>	Điền trung vị cột <i>Fill the column median</i>	Chịu được ngoại lai <i>Resists outliers</i>	Vẫn thu hẹp độ phân tán <i>Still shrinks the spread</i>
Theo nhóm <i>By group</i>	Điền trung bình của nhóm tương tự <i>Fill the mean of a similar group</i>	Sát thực tế hơn <i>Closer to reality</i>	Nhóm nhỏ thì không ổn định <i>Unstable when the group is small</i>
Dự đoán <i>Prediction</i>	Hồi quy hoặc kNN từ cột khác <i>Regression or kNN from other columns</i>	Dùng được quan hệ giữa các cột <i>Uses the relations between columns</i>	Tốn công; có thể làm tương quan đẹp giả <i>Costly; can inflate correlations</i>

Thu nhập của KH07 (hạng thẻ Vàng, nghề kinh doanh tự do, dư nợ 40 triệu đồng) bị thiếu. Mười một khách còn lại có thu nhập 9, 12, 15, 22, 8, 18, 14, 11, 16, 20 và 120 triệu đồng. Đoạn mã tính giá trị điền theo từng cách. Hồi quy dùng 10 khách, bỏ KH12 để một điểm ngoại lai không kéo đường thẳng.

The income of KH07 (Vàng tier, self-employed, balance 40 million dong) is missing. The other eleven customers earn 9, 12, 15, 22, 8, 18, 14, 11, 16, 20 and 120 million dong. The code computes the fill value under each method. The regression uses 10 customers and leaves out KH12 so that one outlying point does not drag the line.

```
import statistics as st

def vn(x):
    return f"{x:.2f}".replace(".", ",")

thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
print("Bỏ dòng thiếu: còn", len(thu_nhap), "trên 12 khách")
print("Điền trung bình:", vn(st.mean(thu_nhap)))
print("Điền trung vị:", vn(st.median(thu_nhap)))
khong_nl = [x for x in thu_nhap if x != 120]
print("Điền trung bình sau khi bỏ 120:", vn(st.mean(khong_nl)))
print("Theo nhóm hạng thẻ Vàng:", vn(st.mean([15, 22, 18, 16])))
print("Theo nhóm nghề kinh doanh tự do:", vn(st.mean([22, 120])))
# hồi quy: thu nhập theo dư nợ, bỏ KH12 (ngoại lai)
x = [12, 18, 20, 45, 10, 30, 16, 14, 25, 35]
y = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20]
mx, my = st.mean(x), st.mean(y)
b = sum((u - mx) * (v - my) for u, v in zip(x, y))
b /= sum((u - mx) ** 2 for u in x)
a0 = my - b * mx
print("Hồi quy: thu nhập =", vn(a0), "+", f"{b:.4f}".replace(".", ","),
      "* dư nợ")
print("Dự đoán cho dư nợ 40:", vn(a0 + b * 40))
```

Kết quả chạy / Output

```
Bỏ dòng thiếu: còn 11 trên 12 khách
Điền trung bình: 24,09
Điền trung vị: 15,00
Điền trung bình sau khi bỏ 120: 14,50
Theo nhóm hạng thẻ Vàng: 17,75
Theo nhóm nghề kinh doanh tự do: 71,00
Hồi quy: thu nhập = 5,57 + 0,3969 * dư nợ
Dự đoán cho dư nợ 40: 21,45
```

Sáu giá trị điền cho KH07 trải từ 14,5 đến 71 triệu đồng. Trung bình 24,09 bị KH12 kéo lên vì mười trong 11 khách kiếm không quá 22 triệu. Trung bình nhóm nghề cho 71 chỉ vì nhóm này gồm KH04 (22) và KH12 (120): nhóm hai người thì không dùng được. Trung vị 15 và trung bình nhóm hạng thẻ Vàng 17,75 sát hơn với 10 khách còn lại, và hồi quy cho 21,45.

The six fill values for KH07 run from 14.5 to 71 million dong. The mean of 24.09 is pulled up by KH12 because ten of the 11 earn at most 22 million. The occupation-group mean gives 71 only because that group holds KH04 (22) and KH12 (120): a two-person group is unusable. The median of 15 and the Vàng-tier mean of 17.75 sit closer to the other 10 customers, and the regression gives 21.45.

Bài mẫu 3.1. Điền giá trị thiếu — *Worked example 3.1. Filling a missing value*

Đề bài: điền thu nhập cho KH07 bằng trung bình, trung vị, trung bình sau khi bỏ ngoại lai và trung bình nhóm hạng thẻ Vàng. Thu nhập 11 khách có giá trị: 8, 9, 11, 12, 14, 15, 16, 18, 20, 22, 120 (sắp tăng dần).

Task: fill the income of KH07 by the mean, the median, the mean after removing the outlier and the mean of the Vàng tier. The 11 known incomes, sorted: 8, 9, 11, 12, 14, 15, 16, 18, 20, 22, 120.

Bước Step	Cách Method	Phép tính Calculation	Kết quả Result
1	Trung bình <i>Mean</i>	$(8 + 9 + 11 + 12 + 14 + 15 + 16 + 18 + 20 + 22 + 120) / 11 = 265 / 11$	24,09
2	Trung vị <i>Median</i>	11 giá trị, số thứ tự $(11 + 1) / 2 = 6$, giá trị thứ 6 <i>11 values, rank $(11 + 1) / 2 = 6$, the 6th value</i>	15
3	Bỏ ngoại lai 120 <i>Without 120</i>	$(265 - 120) / 10 = 145 / 10$	14,5
4	Nhóm hạng thẻ Vàng <i>Vàng tier</i>	KH03, KH04, KH06, KH10: $(15 + 22 + 18 + 16) / 4 = 71 / 4$	17,75

LỖI HAY GẶP / COMMON MISTAKE — Điền trung bình cho cột có ngoại lai. Với 11 giá trị trên, trung bình 24,09 cao hơn 10 trong 11 khách. Đề thi thường cho một giá trị cực đoan để kiểm tra bạn có chuyển sang trung vị (15) hay không.

Filling the mean in a column that has an outlier. With the 11 values above, the mean of 24.09 is higher than 10 of the 11 customers. Exams often include one extreme value to see whether you switch to the median (15).

3.3. Làm sạch: nhiễu, chia giỏ và làm trơn — *3.3. Cleaning: noise, binning and smoothing*

Nhiễu (noise) là sai số ngẫu nhiên trong giá trị đo, ví dụ thu nhập khai làm tròn, dư nợ lệch vài trăm nghìn do chốt khác ngày. Chia giỏ (binning) làm trơn nhiễu bằng cách sắp giá trị tăng dần, chia thành các giỏ, rồi thay mỗi giá trị bằng một con số đại diện của giỏ.

Có hai cách chia. Giỏ đều độ rộng (equal-width) chia khoảng từ giá trị nhỏ nhất đến lớn nhất thành k đoạn bằng nhau. Giỏ đều tần số (equal-frequency) chia sao cho mỗi giỏ có số phần tử gần bằng nhau.

Noise is random error in a measured value, for example income declared in round figures or a balance off by a few hundred thousand because of a different cut-off day. Binning smooths noise by sorting the values, splitting them into bins and replacing each value with a representative figure of its bin. There are two ways to split. Equal-width bins cut the range from the smallest to the largest value into k equal segments. Equal-frequency bins are cut so that each holds a similar number of elements.

$$w = (\max - \min) / k; \text{ biên giỏ } i = \min + i * w \quad (3.1)$$

Với 11 khách có thu nhập, $\min = 8$, $\max = 120$, $k = 3$ thì $w = (120 - 8) / 3 = 37,33$ và biên giỏ là 8; 45,33; 82,67; 120. Giỏ đều tần số trong sách này theo quy ước: 11 giá trị chia 3 giỏ, mỗi giỏ nhận $11 // 3 = 3$ phần tử, giỏ cuối nhận phần dư nên có 5 (3, 3, 5). Quy ước khác (ví dụ 4, 4, 3) cho đáp án khác; đề thi cần ghi rõ.

With the 11 customers' incomes, $\min = 8$, $\max = 120$, $k = 3$, so $w = (120 - 8) / 3 = 37.33$ and the bin edges are 8; 45.33; 82.67; 120. Equal-frequency bins in this book follow one convention: 11 values into 3 bins, each bin takes $11 // 3 = 3$ elements and the last bin takes the remainder, so it holds 5 (3, 3, 5). Another convention (say 4, 4, 3) gives different answers; an exam must state which one applies.

Làm trơn theo trung bình giỏ thay mọi giá trị bằng trung bình của giỏ. Làm trơn theo trung vị dùng trung vị. Làm trơn theo biên giữ hai đầu mút của giỏ và đưa mỗi giá trị về đầu mút gần hơn.

Smoothing by bin means replaces every value with the bin mean. Smoothing by bin medians uses the median. Smoothing by bin boundaries keeps the two end values of a bin and moves each value to the nearer end.

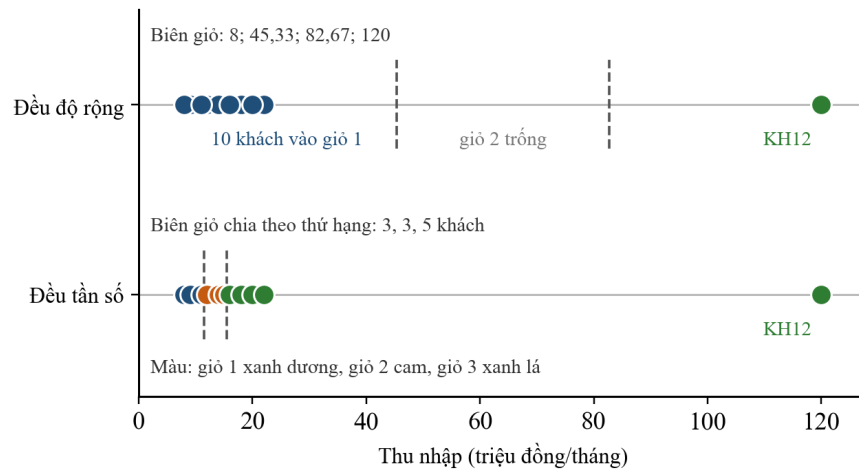
```
import statistics as st

def vn(x):
    return f"{x:.2f}".replace(".", ",")

thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
k = 3
lo, hi = min(thu_nhap), max(thu_nhap)
rong = (hi - lo) / k
print("Biên đều độ rộng:", [vn(lo + i * rong) for i in range(k + 1)])
dr = [min(int((x - lo) / rong), k - 1) for x in thu_nhap]
print("Số phần tử mỗi giỏ:", [dr.count(b) for b in range(k)])
xep = sorted(thu_nhap)
m = len(xep) // k
gio = [xep[i * m:(i + 1) * m] for i in range(k - 1)]
gio.append(xep[(k - 1) * m:])
for i, g in enumerate(gio, 1):
    bien = [g[0] if x - g[0] <= g[-1] - x else g[-1] for x in g]
    print(f"Giỏ {i}: {g}")
    print("    trung bình", vn(st.mean(g)), "| trung vị", st.median(g),
          "| biên", bien)
```

Kết quả chạy / Output

```
Biên đều độ rộng: ['8,00', '45,33', '82,67', '120,00']
Số phần tử mỗi giỏ: [10, 0, 1]
Giỏ 1: [8, 9, 11]
    trung bình 9,33 | trung vị 9 | biên [8, 8, 11]
Giỏ 2: [12, 14, 15]
    trung bình 13,67 | trung vị 14 | biên [12, 15, 15]
Giỏ 3: [16, 18, 20, 22, 120]
    trung bình 39,20 | trung vị 20 | biên [16, 16, 16, 16, 120]
```



Hình 3.2. Thu nhập 11 khách trong giỏ đều độ rộng và đều tần số

Figure 3.2. Income of the 11 customers in equal-width and equal-frequency bins

Giỏ đều độ rộng bị KH12 phá: 10 khách dồn vào giỏ 1, giỏ 2 trống, giỏ 3 chỉ có một người (Hình 3.2). Làm trơn theo trung bình biến mọi giá trị của giỏ 1 thành 14,5, xóa sạch khác biệt giữa người kiếm 8 và người kiếm 22. Giỏ đều tần số tránh được giỏ trống nhưng giỏ 3 vẫn chứa 120: trung bình giỏ là 39,2, cao hơn bốn khách còn lại của giỏ (16 đến 22), còn trung vị 20 thì sát.

Equal-width bins are broken by KH12: 10 customers pile into bin 1, bin 2 is empty and bin 3 holds one person (Figure 3.2). Smoothing by means turns every value of bin 1 into 14.5, wiping out the difference between someone earning 8 and someone earning 22. Equal-frequency bins avoid the empty bin, but bin 3 still contains 120: its mean is 39.2, above the other four customers of that bin (16 to 22), while the median of 20 stays close.

Bài mẫu 3.2. Làm trơn giỏ đều tần số — *Worked example 3.2. Smoothing equal-frequency bins*

Đề bài: chia 11 thu nhập thành 3 giỏ đều tần số (3, 3, 5), làm trơn theo trung bình, trung vị và biên giỏ.

Task: split the 11 incomes into 3 equal-frequency bins (3, 3, 5) and smooth by bin means, medians and boundaries.

Giỏ Bìn	Giá trị Values	Trung bình Mean	Trung vị Median	Theo biên Boundaries
1	8, 9, 11	$28 / 3 = 9,33$	9	8, 8, 11
2	12, 14, 15	$41 / 3 = 13,67$	14	12, 15, 15
3	16, 18, 20, 22, 120	$196 / 5 = 39,2$	20	16, 16, 16, 16, 120

Cách làm theo biên: giỏ 2 có hai đầu mút 12 và 15; giá trị 14 cách 12 là 2, cách 15 là 1 nên chuyển thành 15. Giỏ 3 có đầu mút 16 và 120; 18, 20, 22 đều gần 16 hơn.

Boundary method: bin 2 has the end values 12 and 15; the value 14 is 2 from 12 and 1 from 15, so it becomes 15. Bin 3 has the end values 16 and 120; 18, 20 and 22 are all nearer to 16.

LỖI HAY GẶP / COMMON MISTAKE — Chia giỏ đều độ rộng khi dữ liệu có giá trị cực đoan rồi kết luận "phân phối thu nhập chỉ có hai nhóm". Biên giỏ do khoảng min–max quyết định, nên một điểm 120 đẩy ba biên giỏ ra xa 10 điểm còn lại.

Using equal-width bins on data with an extreme value and concluding that "income has only two groups". The bin edges follow the min–max range, so one point at 120 pushes the edges far from the other 10 points.

3.4. Ngoại lai: phát hiện theo IQR và z-score — 3.4. Outliers: detection by IQR and z-score

Ngoại lai (outlier) là giá trị lệch xa phần lớn dữ liệu. Ngoại lai có thể là lỗi nhập, có thể là trường hợp hiếm nhưng có thật. Nhiều và ngoại lai hay bị nhầm, nên bảng sau đặt cạnh nhau.

An outlier is a value far from most of the data. It may be an entry error or a rare but genuine case. Noise and outliers are often confused, so the table puts them side by side.

Tiêu chí Criterion	Nhiều Noise	Ngoại lai Outlier
Bản chất Nature	Sai số ngẫu nhiên nhỏ phủ lên mọi giá trị <i>Small random error laid over every value</i>	Một vài giá trị lệch xa, có khi là thật <i>A few distant values, sometimes genuine</i>
Phạm vi Scope	Ảnh hưởng nhiều quan sát cùng lúc <i>Affects many observations at once</i>	Chỉ ít quan sát <i>Only a few observations</i>
Xử lý chính Main treatment	Làm trơn (giỏ, trung bình trượt) <i>Smoothing (bins, moving average)</i>	Kiểm nguồn, giữ, cắt đầu mút hoặc tách riêng <i>Check the source, keep, winsorise or model separately</i>
Có thể là thông tin? Can it be signal?	Thường không <i>Usually not</i>	Có: gian lận, khách giàu bất thường <i>Yes: fraud, an unusually rich customer</i>

Quy tắc IQR (khoảng tứ phân vị) coi x là ngoại lai nếu nằm ngoài hai hàng rào. Quy ước tứ phân vị trong sách là phương pháp inclusive của hàm quantiles trong thư viện chuẩn Python: vị trí của phân vị p trong n giá trị đã sắp là $(n - 1) * p + 1$, nội suy tuyến tính giữa hai giá trị kề.

The IQR (interquartile range) rule treats x as an outlier if it lies outside the two fences. The quartile convention in this book is the inclusive method of the quantiles function in the Python standard library: the position of percentile p among n sorted values is $(n - 1) * p + 1$, with linear interpolation between the two neighbouring values.

$$\text{IQR} = Q3 - Q1; \text{ ngoại lai nếu } x < Q1 - 1,5 * \text{IQR} \text{ hoặc } x > Q3 + 1,5 * \text{IQR} \quad (3.2)$$

$$z = (x - \bar{x}) / s \quad (3.3)$$

Quy tắc z-score coi x là ngoại lai khi $|z| > 3$ (đôi khi dùng 2,5). s là độ lệch chuẩn mẫu, mẫu số $n - 1$, theo quy ước toàn sách.

The z-score rule treats x as an outlier when $|z| > 3$ (sometimes 2.5). Here s is the sample standard deviation with denominator $n - 1$, the convention used throughout the book.

```
import statistics as st

def vn(x, k=4):
    return f"{x:.{k}f}".replace(".", ",")

tn = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
q1, q2, q3 = st.quantiles(tn, n=4, method="inclusive")
iqr = q3 - q1
duoi, tren = q1 - 1.5 * iqr, q3 + 1.5 * iqr
print("Q1 =", vn(q1, 2), "| Q3 =", vn(q3, 2), "| IQR =", vn(iqr, 2))
print("Hàng rào:", vn(duoi, 2), "đến", vn(tren, 2))
nl = [x for x in tn if x < duoi or x > tren]
print("Ngoại lai theo IQR:", nl)
tb, s = st.mean(tn), st.stdev(tn)
z = [(x - tb) / s for x in tn]
print("Trung bình", vn(tb, 2), "| độ lệch chuẩn mẫu", vn(s, 2))
print("z của 120:", vn(max(z)), "| có |z| > 3:", any(abs(v) > 3 for v in z))
sach = [x for x in tn if x not in nl]
tb2, s2 = st.mean(sach), st.stdev(sach)
print("z của 120 so với 10 khách còn lại:", vn((120 - tb2) / s2))
```

Kết quả chạy / Output

```
Q1 = 11,50 | Q3 = 19,00 | IQR = 7,50
Hàng rào: 0,25 đến 30,25
Ngoại lai theo IQR: [120]
Trung bình 24,09 | độ lệch chuẩn mẫu 32,11
z của 120: 2,9868 | có |z| > 3: False
z của 120 so với 10 khách còn lại: 22,8117
```

Hai quy tắc cho hai kết luận. IQR bắt được 120 vì hàng rào trên là 30,25. Z-score của 120 là 2,9868, nhỏ hơn 3, nên quy tắc $|z| > 3$ bỏ sót. Lý do: KH12 tự kéo trung bình lên 24,09 và độ lệch chuẩn lên 32,11. Khi so với 10 khách còn lại (trung bình 14,5), z của 120 là 22,8117. Mẫu nhỏ và ngoại lai nặng khiến z-score bị che khuất; IQR dùng phân vị nên ít chịu ảnh hưởng.

The two rules reach different conclusions. The IQR rule catches 120 because the upper fence is 30.25. The z-score of 120 is 2.9868, less than 3, so the $|z| > 3$ rule misses it. The reason: KH12

itself lifts the mean to 24.09 and the standard deviation to 32.11. Against the other 10 customers (mean 14.5), the z of 120 is 22.8117. A small sample with a heavy outlier masks the z -score; the IQR uses percentiles and is far less affected.

Bài mẫu 3.3. Tìm ngoại lai bằng IQR — Worked example 3.3. Finding outliers with the IQR

Bước Step	Việc làm Action	Phép tính Calculation	Kết quả Result
1	Sắp tăng dần Sort	8, 9, 11, 12, 14, 15, 16, 18, 20, 22, 120 ($n = 11$)	
2	Vị trí Q1 Position of $Q1$	$(11 - 1) * 0,25 + 1 = 3,5$: giữa giá trị thứ 3 (11) và thứ 4 (12) $(11 - 1) * 0.25 + 1 = 3.5$: between the 3rd value (11) and the 4th (12)	$Q1 = 11,5$
3	Vị trí Q3 Position of $Q3$	$(11 - 1) * 0,75 + 1 = 8,5$: giữa giá trị thứ 8 (18) và thứ 9 (20) $(11 - 1) * 0.75 + 1 = 8.5$: between the 8th value (18) and the 9th (20)	$Q3 = 19$
4	IQR	$19 - 11,5$	7,5
5	Hàng rào Fences	$11,5 - 1,5 * 7,5$ và $19 + 1,5 * 7,5$	0,25; 30,25
6	Kết luận Conclusion	$120 > 30,25$; các giá trị còn lại nằm trong $[0,25; 30,25]$ $120 > 30.25$; the rest lie inside $[0.25; 30.25]$	Ngoại lai: 120 Outlier: 120

Việc làm với KH12 phụ thuộc nguồn gốc. Hồ sơ cho thấy nghề kinh doanh tự do, hạng thẻ Bạch kim, dư nợ 150 triệu đồng: tỷ lệ dư nợ trên thu nhập là 1,25, ngang các khách khác. Số liệu nhất quán nên giữ dòng này, đánh dấu cờ ngoại lai, và dùng trung vị, giỏ đều tần số hoặc chuẩn hóa chịu được ngoại lai ở bước sau. Nếu kiểm nguồn thấy gõ thừa số 0 (12 thành 120) thì sửa giá trị.

What to do with KH12 depends on its origin. The file shows self-employed, Bạch kim tier and a balance of 150 million dong: the balance-to-income ratio is 1.25, in line with the others. The figures are consistent, so keep the row, set an outlier flag, and use the median, equal-frequency bins or outlier-resistant scaling later. If the source check finds an extra zero typed (12 became 120), correct the value.

PHẢI THUỘC / MEMORISE — IQR dùng phân vị nên bắt được ngoại lai mà không bị ngoại lai làm lệch ngưỡng. Z-score dùng trung bình và độ lệch chuẩn, hai số bị chính ngoại lai kéo, nên với mẫu nhỏ có thể bỏ sót (z của 120 là $2,9868 < 3$).

The IQR uses percentiles and so catches outliers without the outlier shifting the threshold. The z -score uses the mean and standard deviation, both pulled by the outlier, so on a small sample it can miss (the z of 120 is $2.9868 < 3$).

3.5. Tích hợp dữ liệu — 3.5. Data integration

Tích hợp dữ liệu (data integration) ghép dữ liệu từ nhiều nguồn thành một bảng. Ba vấn đề hay gặp được tóm trong bảng sau.

Data integration merges data from several sources into one table. Three common problems are summarised in the table.

Vấn đề <i>Problem</i>	Biểu hiện <i>Symptom</i>	Cách xử lý <i>Remedy</i>
Trùng thực thể <i>Entity resolution</i>	Một khách có hai mã: một mã ở hệ thống cho vay, một mã ở ứng dụng <i>One customer holds two codes: one in the lending system, one in the app</i>	Ghép theo khóa chung đã ẩn danh hóa; nếu không có thì ghép theo nhiều trường và duyệt tay cặp nghi ngờ <i>Match on a shared pseudonymised key; failing that, on several fields with manual review of doubtful pairs</i>
Xung đột giá trị <i>Value conflict</i>	Thu nhập tính bằng đồng ở nguồn này, triệu đồng ở nguồn kia; ngày viết 03/04 hay 04/03 <i>Income in dong in one source, million dong in another; a date written 03/04 or 04/03</i>	Thống nhất đơn vị và định dạng trước khi gộp; ghi đơn vị trong tên cột <i>Fix units and formats before merging; put the unit in the column name</i>
Thuộc tính dư thừa <i>Redundant attribute</i>	Cột này suy ra được từ cột khác (tuổi và ngày sinh) hoặc hai cột tương quan gần 1 <i>One column derivable from another (age and birth date) or two highly correlated columns</i>	Đo tương quan Pearson cho thuộc tính số, kiểm định χ^2 cho thuộc tính định danh; giữ một cột <i>Pearson correlation for numeric attributes, the χ^2 test for nominal ones; keep one column</i>

Hệ số tương quan Pearson r của hai thuộc tính số nằm trong $[-1; 1]$; $|r|$ gần 1 báo hiệu thuộc tính này gần như suy ra được từ thuộc tính kia. Giá trị r của thu nhập và dư nợ ở Chương 2 là 0,9898 trên 11 khách.

The Pearson coefficient r of two numeric attributes lies in $[-1; 1]$; $|r|$ near 1 signals that one attribute is almost derivable from the other. The r of income and balance in Chapter 2 is 0.9898 over 11 customers.

```
import math

def pearson(x, y):
    mx, my = sum(x) / len(x), sum(y) / len(y)
    tu = sum((a - mx) * (b - my) for a, b in zip(x, y))
    mau = math.sqrt(sum((a - mx) ** 2 for a in x) *
                    sum((b - my) ** 2 for b in y))
    return tu / mau

thu_nhap = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
du_no = [12, 18, 20, 45, 10, 30, 16, 14, 25, 35, 150]
r = pearson(thu_nhap, du_no)
r2 = pearson(thu_nhap[:-1], du_no[:-1])
print(f"r (11 khách, có KH12): {r:.4f}".replace(".", ","))
print(f"r (10 khách, bỏ KH12): {r2:.4f}".replace(".", ","))
print("Nghỉ dư thừa nếu |r| > 0,9:", abs(r) > 0.9, abs(r2) > 0.9)
```

Kết quả chạy / Output

```
r (11 khách, có KH12): 0,9898
r (10 khách, bỏ KH12): 0,9627
Nghỉ dư thừa nếu |r| > 0,9: True True
```

$r = 0,9898$ khi có KH12 và $0,9627$ khi bỏ KH12. Cả hai đều vượt $0,9$, nhưng con số đầu bị một điểm cực đoan đẩy lên. Kết luận "thu nhập và dư nợ gần như cùng mang một thông tin" đúng ở cả hai trường hợp, song không nên bỏ cột chỉ vì r cao: hai cột này có ý nghĩa nghiệp vụ khác nhau (khả năng trả và nghĩa vụ phải trả), và tỷ lệ giữa chúng, DTI ở mục 3.7, mang thông tin mà từng cột riêng không có.

$r = 0.9898$ with KH12 and 0.9627 without. Both exceed 0.9 , but the first figure is pushed up by one extreme point. The conclusion "income and balance carry nearly the same information" holds either way, yet a column should not be dropped on a high r alone: the two have different business meanings (ability to repay and obligation to repay), and the ratio between them, DTI in section 3.7, carries information that neither column holds separately.

Với hai thuộc tính định danh, kiểm định χ^2 (chi-bình phương) xét hai thuộc tính có độc lập hay không. Bảng 2×2 sau là dữ kiện đề bài: 200 khách, thuộc tính dùng ứng dụng Phú Sa (có, không) và thuộc tính từng trễ hạn (có, không). Công thức có kỳ vọng suy từ tổng hàng và tổng cột.

For two nominal attributes, the χ^2 (chi-squared) test asks whether they are independent. The 2×2 table is given in the task: 200 customers, with the attribute uses the Phú Sa app (yes, no) and the attribute has been late (yes, no). The formula uses expected counts derived from row and column totals.

$$\chi^2 = \sum (O - E)^2 / E; E = (\text{tổng hàng} * \text{tổng cột}) / \text{tổng chung} \quad (3.4)$$

Dữ kiện <i>Given</i>	Trễ hạn <i>Late</i>	Không trễ <i>Not late</i>	Cộng <i>Total</i>
Dùng ứng dụng <i>Uses the app</i>	18	102	120
Không dùng <i>Does not use</i>	22	58	80
Cộng <i>Total</i>	40	160	200

```
# hàng: dùng ứng dụng / không dùng; cột: có trễ hạn / không trễ hạn
bang = [[18, 102],
         [22, 58]]
n = sum(sum(h) for h in bang)
hang = [sum(h) for h in bang]
cot = [sum(c) for c in zip(*bang)]
chi2 = 0
for i in range(2):
    for j in range(2):
        kv = hang[i] * cot[j] / n
        chi2 += (bang[i][j] - kv) ** 2 / kv
        print(f"Ô ({i + 1}, {j + 1}): quan sát {bang[i][j]:>3}, "
              f"kỳ vọng {kv:.2f}".replace(".", ","))
print("χ² =", f"{chi2:.4f}".replace(".", ","), "| bậc tự do 1")
print("Ngưỡng 5% là 3,841; bác bỏ độc lập:", chi2 > 3.841)
```

Kết quả chạy / Output

```

Ô (1,1): quan sát 18, kỳ vọng 24,00
Ô (1,2): quan sát 102, kỳ vọng 96,00
Ô (2,1): quan sát 22, kỳ vọng 16,00
Ô (2,2): quan sát 58, kỳ vọng 64,00
 $\chi^2 = 4,6875$  | bậc tự do 1
Ngưỡng 5% là 3,841; bác bỏ độc lập: True

```

Bài mẫu 3.4. Kiểm định χ^2 cho bảng 2×2 — *Worked example 3.4. The χ^2 test on a 2×2 table*

Bước Step	Ô Cell	Kỳ vọng E Expected E	$(O - E)^2 / E$
1	Dùng, trễ (O = 18) <i>Uses, late (O = 18)</i>	$120 * 40 / 200 = 24$	$(18 - 24)^2 / 24 = 1,5$
2	Dùng, không trễ (O = 102) <i>Uses, not late (O = 102)</i>	$120 * 160 / 200 = 96$	$(102 - 96)^2 / 96 = 0,375$
3	Không dùng, trễ (O = 22) <i>Does not use, late (O = 22)</i>	$80 * 40 / 200 = 16$	$(22 - 16)^2 / 16 = 2,25$
4	Không dùng, không trễ (O = 58) <i>Does not use, not late (O = 58)</i>	$80 * 160 / 200 = 64$	$(58 - 64)^2 / 64 = 0,5625$
5	Cộng <i>Sum</i>	bậc tự do $(2 - 1)(2 - 1) = 1$ <i>degrees of freedom (2 - 1)(2 - 1) = 1</i>	$\chi^2 = 4,6875$

So với ngưỡng 3,841 (bậc tự do 1, mức ý nghĩa 5%), 4,6875 lớn hơn nên bác bỏ giả thuyết độc lập: dùng ứng dụng và từng trễ hạn có liên hệ thống kê. Khách dùng ứng dụng trễ hạn 18 trên 120 (15%), khách không dùng 22 trên 80 (27,5%). Kiểm định chỉ nói có liên hệ, không nói ứng dụng làm giảm trễ hạn.

Against the threshold of 3.841 (1 degree of freedom, 5% level), 4.6875 is larger, so independence is rejected: using the app and having been late are statistically related. App users were late 18 times in 120 (15%), non-users 22 times in 80 (27.5%). The test says only that a relation exists, not that the app lowers lateness.

LỖI HAY GẶP / COMMON MISTAKE — Tính kỳ vọng bằng cách chia đều ($200 / 4 = 50$ mỗi ô). Kỳ vọng phải theo tổng hàng và tổng cột: ô đầu là $120 * 40 / 200 = 24$, không phải 50. *Computing expected counts by splitting evenly ($200 / 4 = 50$ per cell). Expected counts follow the row and column totals: the first cell is $120 * 40 / 200 = 24$, not 50.*

3.6. Biến đổi: chuẩn hóa dữ liệu — 3.6. Transformation: normalisation

Tuổi nằm trong 23 đến 50, thu nhập trong 8 đến 120 triệu đồng, dư nợ trong 10 đến 150 triệu đồng. Khi tính khoảng cách Euclid như ở Chương 2, cột có biên độ lớn chi phối tổng, hai cột còn lại gần như không đóng góp. Chuẩn hóa (normalisation) đưa các cột về

thang đo cùng cỡ. Ba cách thường dùng được viết dưới dạng công thức; z-score là công thức (3.3).

Age lies between 23 and 50, income between 8 and 120 million dong, balance between 10 and 150 million dong. When Euclidean distance is computed as in Chapter 2, the column with the widest range dominates the sum and the other two hardly contribute. Normalisation puts the columns on scales of the same size. Three common methods are written as formulas; the z-score is formula (3.3).

$$x' = (x - \min) / (\max - \min) * (\max \text{ mới} - \min \text{ mới}) + \min \text{ mới} \quad (3.5)$$

$$x' = x / 10^j, j \text{ là số nguyên nhỏ nhất sao cho } \max |x'| < 1 \quad (3.6)$$

Min-max co giãn tuyến tính về đoạn [min mới, max mới], thường là [0, 1]. Z-score trừ trung bình rồi chia độ lệch chuẩn mẫu, cho cột trung bình 0 và độ lệch chuẩn 1. Chuẩn hóa thập phân (decimal scaling) chia cho lũy thừa của 10. Trong thư viện scikit-learn, ba cách đều có MinMaxScaler và StandardScaler; chuẩn hóa thập phân thường tự viết.

Min-max stretches linearly onto [new min, new max], usually [0, 1]. The z-score subtracts the mean and divides by the sample standard deviation, giving a column with mean 0 and standard deviation 1. Decimal scaling divides by a power of 10. In scikit-learn the first two have MinMaxScaler and StandardScaler; decimal scaling is usually hand-written.

```
import statistics as st

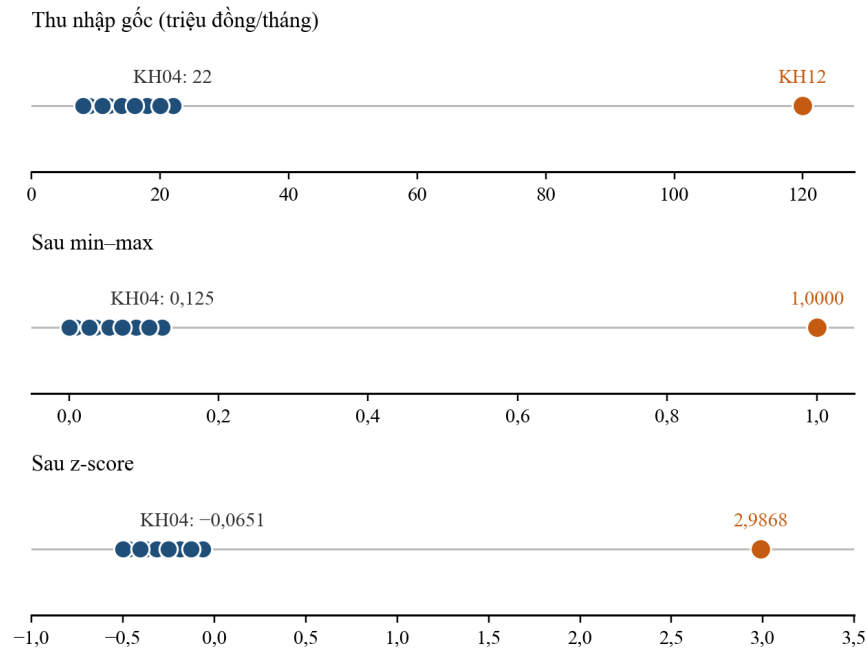
def vn(x, k=4):
    return f"{x:.{k}f}".replace(".", ",")

tn = [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120]
lo, hi = min(tn), max(tn)
tb, s = st.mean(tn), st.stdev(tn)
j = 0
while max(abs(x) for x in tn) / 10 ** j >= 1:
    j += 1
print("Khách Thu nhập Min-max Z-score Thập phân")
for ma, x in [("KH01", 9), ("KH04", 22), ("KH05", 8), ("KH12", 120)]:
    mm = (x - lo) / (hi - lo)
    z = (x - tb) / s
    print(f"{ma:<7}{x:>8}{vn(mm):>10}{vn(z):>9}{vn(x / 10 ** j, 3):>11}")
print("j =", j, "| 10^j =", 10 ** j)
```

Kết quả chạy / Output

Khách	Thu nhập	Min-max	Z-score	Thập phân
KH01	9	0,0089	-0,4700	0,009
KH04	22	0,1250	-0,0651	0,022
KH05	8	0,0000	-0,5011	0,008
KH12	120	1,0000	2,9868	0,120

j = 3 | 10^j = 1000



Hình 3.3. Thu nhập 11 khách trước và sau chuẩn hóa: hình dạng giữ nguyên, thang đo đổi
Figure 3.3. Income of the 11 customers before and after normalisation: the shape stays, the scale changes

Cả hai cách cho cùng hình dạng phân phối (Hình 3.3). KH12 bị đưa về 1,0000 theo min-max và 2,9868 theo z-score, còn mười khách kia bị nén vào đoạn $[0; 0,125]$ theo min-max và đoạn $[-0,5011; -0,0651]$ theo z-score. Chuẩn hóa không làm ngoại lai biến mất; nó chỉ đổi thang đo.

Both methods leave the shape of the distribution as it was (Figure 3.3). KH12 goes to 1.0000 under min-max and 2.9868 under the z-score, while the other ten customers are squeezed into $[0; 0.125]$ under min-max and into $[-0.5011; -0.0651]$ under the z-score. Normalisation does not make an outlier disappear; it only changes the scale.

Tiêu chí <i>Criterion</i>	Min-max	Z-score	Thập phân <i>Decimal scaling</i>
Kết quả <i>Output</i>	Trong $[0, 1]$ khi min, max lấy từ chính dữ liệu <i>Within $[0, 1]$ when min and max come from the same data</i>	Trung bình 0, độ lệch chuẩn 1; không bị chặn <i>Mean 0, standard deviation 1; unbounded</i>	Trong $(-1, 1)$
Nhạy ngoại lai <i>Outlier sensitivity</i>	Cao: min hoặc max đổi thì mọi giá trị đổi <i>High: if min or max changes, every value changes</i>	Vừa: trung bình và độ lệch chuẩn bị kéo <i>Moderate: mean and deviation are pulled</i>	Thấp, nhưng chỉ cắt thang <i>Low, but only rescales</i>
Dùng khi <i>Use when</i>	Cần thang cố định (mạng nơ-ron, kNN) <i>A fixed scale is needed (neural nets, kNN)</i>	Dữ liệu gần chuông, thuật toán giả định chuẩn <i>Data near bell-shaped, algorithm assumes normality</i>	Chỉ cần đưa số lớn về cỡ nhỏ <i>Only need to shrink large numbers</i>

Bài mẫu 3.5. Chuẩn hóa thu nhập của KH04 — *Worked example 3.5. Normalising the income of KH04*

Đề bài: thu nhập 11 khách như ở mục 3.2; chuẩn hóa thu nhập 22 của KH04 theo ba cách. Cho tổng 265, trung bình 24,09, độ lệch chuẩn mẫu 32,11.

Task: the 11 incomes as in section 3.2; normalise the income of 22 for KH04 by the three methods. Given sum 265, mean 24.09, sample standard deviation 32.11.

Bước Step	Cách Method	Phép tính Calculation	Kết quả Result
1	Min-max	$\min = 8, \max = 120: (22 - 8) / (120 - 8) = 14 / 112$	0,125
2	Z-score	$(22 - 24,09) / 32,11 = -2,09 / 32,11$	-0,0651
3	Thập phân <i>Decimal</i>	$\max = 120$ nên $10^2 = 100$ chưa đủ ($1,2 \geq 1$), $10^3 = 1.000$: $22 / 1.000$ <i>max = 120 so $10^2 = 100$ is too small ($1.2 \geq 1$), $10^3 = 1,000$: $22 / 1,000$</i>	0,022
4	Kiểm tra <i>Check</i>	Thu nhập 8 theo min-max <i>Income 8 under min-max</i>	0

LỖI HAY GẶP / COMMON MISTAKE — Dùng độ lệch chuẩn tổng thể (chia n) thay vì mẫu (chia n - 1) khi đề bài là mẫu. Với 11 giá trị trên, s = 32,11 (chia 10) còn 30,62 (chia 11): z của KH04 đổi từ -0,0651 sang -0,0683 và đáp án sai chữ số thứ ba.

Using the population standard deviation (divide by n) instead of the sample one (divide by n - 1) when the task is a sample. With the 11 values above, s = 32.11 (divide by 10) against 30.62 (divide by 11): the z of KH04 moves from -0.0651 to -0.0683 and the answer is wrong from the third digit.

3.7. Biến đổi: mã hóa, rời rạc hóa và thuộc tính mới — 3.7. Transformation: encoding, discretisation and new attributes

Hầu hết thuật toán tính toán trên số, nên thuộc tính chữ phải được mã hóa. Cách mã hóa phụ thuộc kiểu thuộc tính ở Chương 2.

Most algorithms compute on numbers, so text attributes must be encoded. The way to encode depends on the attribute type from Chapter 2.

Kiểu thuộc tính <i>Attribute type</i>	Cách mã hóa <i>Encoding</i>	Ví dụ ở Phú Sa <i>Example at Phú Sa</i>
Nhị phân <i>Binary</i>	0 và 1 <i>0 and 1</i>	Có nhà: 1; không có nhà: 0 <i>Owns a home: 1; does not: 0</i>
Định danh <i>Nominal</i>	One-hot: mỗi giá trị một cột 0/1; có thể bỏ một cột (dummy) <i>One-hot: one 0/1 column per value; one column may be dropped (dummy)</i>	Nghề: công nhân, nhân viên văn phòng, giáo viên, kinh doanh tự do <i>Occupation: worker, office staff, teacher, self-employed</i>
Thứ bậc <i>Ordinal</i>	Số thứ hạng, có thể đưa về [0, 1] <i>Rank numbers, optionally rescaled to [0, 1]</i>	Hạng thẻ: Chuẩn 1, Vàng 2, Bạch kim 3 <i>Card tier: Chuẩn 1, Vàng 2, Bạch kim 3</i>

```

nghe = ["Công nhân", "Nhân viên văn phòng", "Giáo viên",
        "Kinh doanh tự do"]
khach = [("KH01", "Công nhân", "Chuẩn"),
         ("KH04", "Kinh doanh tự do", "Vàng"),
         ("KH11", "Giáo viên", "Bạch kim")]
thu_tu = {"Chuẩn": 1, "Vàng": 2, "Bạch kim": 3}
print("One-hot (4 cột) và dummy (bỏ cột đầu, 3 cột):")
for ma, n, _ in khach:
    oh = [1 if n == g else 0 for g in nghe]
    print(f" {ma}: {oh} -> {oh[1:]}")
print("Mã hóa thứ bậc rồi đưa về [0, 1]:")
for ma, _, h in khach:
    r = thu_tu[h]
    chuan = (r - 1) / (len(thu_tu) - 1)
    print(f" {ma}: {h:<9} -> {r} -> {chuan:.1f}".replace(".", ","))

```

Kết quả chạy / Output

```

One-hot (4 cột) và dummy (bỏ cột đầu, 3 cột):
KH01: [1, 0, 0, 0] -> [0, 0, 0]
KH04: [0, 0, 0, 1] -> [0, 0, 1]
KH11: [0, 0, 1, 0] -> [0, 1, 0]
Mã hóa thứ bậc rồi đưa về [0, 1]:
KH01: Chuẩn      -> 1 -> 0,0
KH04: Vàng       -> 2 -> 0,5
KH11: Bạch kim   -> 3 -> 1,0

```

One-hot cho nghề tạo 4 cột. Cột thứ tư suy ra được từ ba cột kia (tổng bốn cột luôn bằng 1), nên mô hình tuyến tính thường bỏ một cột để tránh cộng tuyến hoàn hảo. Hạng thẻ giữ thứ tự nên mã 1, 2, 3 hợp lý, và chuẩn hóa $(r - 1) / (3 - 1)$ cho 0; 0,5; 1. Điều không được làm là mã hóa nghề thành 1, 2, 3, 4: mô hình sẽ hiểu giáo viên "lớn hơn" công nhân và "cách" kinh doanh tự do một đơn vị.

One-hot for occupation creates 4 columns. The fourth can be derived from the other three (the four always sum to 1), so linear models usually drop one to avoid perfect collinearity. Card tier keeps an order, so codes 1, 2, 3 are reasonable, and rescaling $(r - 1) / (3 - 1)$ gives 0; 0.5; 1. What must not be done is to code occupation as 1, 2, 3, 4: the model would read teacher as "greater than" worker and one unit "away" from self-employed.

Rời rạc hóa (discretisation) đổi thuộc tính số thành các nhóm. Có thể chia theo độ rộng, theo tần số như ở mục 3.3, hoặc theo ngưỡng nghiệp vụ. Xây dựng thuộc tính mới

(feature construction) tạo cột từ cột có sẵn. Ở Phú Sa, tỷ lệ DTI đo dư nợ so với thu nhập tháng.

Discretisation turns a numeric attribute into groups. The split may be by width, by frequency as in section 3.3, or by business threshold. Feature construction creates a column from existing ones. At Phú Sa, the DTI ratio measures balance against monthly income.

$$\text{DTI} = \text{dư nợ} / \text{thu nhập tháng} \quad (3.7)$$

Cách tính ở đây là dạng đơn giản hóa (dư nợ tính bằng số tháng thu nhập). Nhiều tổ chức tín dụng dùng nghĩa vụ trả nợ hằng tháng chia thu nhập hằng tháng; đề thi sẽ cho công thức cần dùng.

The calculation here is a simplified form (balance measured in months of income). Many credit institutions use the monthly debt obligation divided by monthly income; an exam will give the formula to apply.

```
tuoi = [25, 29, 34, 41, 23, 38, 45, 31, 27, 36, 50, 47]
thu_nhap = [9, 12, 15, 22, 8, 18, None, 14, 11, 16, 20, 120]
du_no = [12, 18, 20, 45, 10, 30, 40, 16, 14, 25, 35, 150]
ma = [f"KH{i:02d}" for i in range(1, 13)]
dti = {m: d / t for m, d, t in zip(ma, du_no, thu_nhap) if t}
for m in ("KH01", "KH04", "KH12"):
    print(f"DTI {m} = {dti[m]:.4f}".replace(".", ","))
cao = max(dti, key=dti.get)
print("DTI cao nhất:", cao, f"({dti[cao]:.2f})".replace(".", ","))
print("Không tính được DTI:", [m for m in ma if m not in dti])
nhom = {"dưới 30": 0, "30 đến 39": 0, "từ 40": 0}
for t in tuoi:
    nhom["dưới 30" if t < 30 else "30 đến 39" if t < 40 else "từ 40"] += 1
print("Rời rạc hóa tuổi:", nhom)
```

Kết quả chạy / Output

```
DTI KH01 = 1,3333
DTI KH04 = 2,0455
DTI KH12 = 1,2500
DTI cao nhất: KH04 (2,05)
Không tính được DTI: ['KH07']
Rời rạc hóa tuổi: {'dưới 30': 4, '30 đến 39': 4, 'từ 40': 4}
```

KH12 có thu nhập ngoại lai nhưng DTI 1,25 nằm giữa các khách khác (KH01 1,33; KH04 2,05): tỷ lệ đã hóa giải thang đo. KH04 có DTI cao nhất, 2,05. KH07 không tính được DTI vì thiếu thu nhập, nên thuộc tính mới thừa hưởng ô thiếu của cột gốc. Ba nhóm tuổi (dưới 30, từ 30 đến 39, từ 40) chia 12 khách thành 4, 4, 4.

KH12 has an outlying income, yet a DTI of 1.25 sits among the other customers (KH01 1.33; KH04 2.05): the ratio has absorbed the scale. KH04 has the highest DTI, 2.05. KH07 has no DTI because income is missing, so the new attribute inherits the blank of its source column. The three age groups (under 30, 30 to 39, 40 and over) split the 12 customers 4, 4, 4.

Bài mẫu 3.6. Mã hóa một dòng dữ liệu — Worked example 3.6. Encoding one row

Đề bài: KH04 làm nghề kinh doanh tự do, hạng thẻ Vàng, thu nhập 22 triệu đồng, dư nợ 45 triệu đồng. Biểu diễn dòng này sau mã hóa.

Task: KH04 is self-employed, holds a Vàng card, earns 22 million dong and owes 45 million dong. Show the row after encoding.

Bước Step	Thuộc tính Attribute	Phép tính Calculation	Giá trị mới New value
1	Nghề <i>Occupation</i>	Thứ tự cột: công nhân, nhân viên văn phòng, giáo viên, kinh doanh tự do <i>Column order: worker, office staff, teacher, self-employed</i>	0, 0, 0, 1
2	Hạng thẻ <i>Card tier</i>	Vàng = 2; $(2 - 1) / (3 - 1)$	0,5
3	DTI	45 / 22	2,0455
4	Dummy <i>Dummy</i>	Bỏ cột đầu (công nhân) <i>Drop the first column (worker)</i>	0, 0, 1

LỖI HAY GẶP / COMMON MISTAKE — Mã hóa thuộc tính định danh bằng số nguyên liên tiếp (công nhân 1, nhân viên văn phòng 2...). Thuật toán dùng khoảng cách hoặc hệ số sẽ hiểu có thứ tự. Dùng one-hot, và chỉ mã số thứ hạng cho thuộc tính thứ bậc như hạng thẻ.
Encoding a nominal attribute with consecutive integers (worker 1, office staff 2...). Algorithms that use distances or coefficients will read an order into it. Use one-hot, and give rank numbers only to ordinal attributes such as card tier.

3.8. Thu giảm số chiều: chọn thuộc tính và PCA — 3.8. Dimensionality reduction: attribute selection and PCA

Thu giảm dữ liệu (data reduction) tạo ra biểu diễn nhỏ hơn mà vẫn cho kết quả khai phá gần như cũ. Có hai hướng: bỏ bớt thuộc tính, hoặc ghép nhiều thuộc tính thành ít thuộc tính mới. Chọn thuộc tính (feature selection) theo ba cách.

Data reduction produces a smaller representation that gives almost the same mining results. There are two directions: drop attributes, or combine many attributes into a few new ones. Feature selection follows three approaches.

Cách Approach	Nguyên tắc Principle	Ví dụ Example	Đặc điểm Trait
Lọc <i>Filter</i>	Chấm điểm từng thuộc tính độc lập với mô hình, giữ điểm cao <i>Score each attribute independent of the model, keep the high scores</i>	Tương quan với nhãn, χ^2 , phương sai <i>Correlation with the label, χ^2, variance</i>	Nhanh; bỏ qua tương tác giữa các cột <i>Fast; ignores interaction between columns</i>
Bao <i>Wrapper</i>	Thử các tập con, đo kết quả mô hình, giữ tập tốt nhất <i>Try subsets, measure the model, keep the best subset</i>	Chọn tiến, loại dần <i>Forward selection, backward elimination</i>	Sát mô hình; tốn thời gian <i>Close to the model; slow</i>

Cách <i>Approach</i>	Nguyên tắc <i>Principle</i>	Ví dụ <i>Example</i>	Đặc điểm <i>Trait</i>
Nhúng <i>Embedded</i>	Việc chọn nằm trong quá trình huấn luyện <i>Selection happens inside training</i>	Độ quan trọng ở cây quyết định, hồi quy phạt L1 <i>Importance in decision trees, L1-penalised regression</i>	Cân bằng chi phí và chất lượng <i>Balances cost and quality</i>

Ví dụ lọc theo tương quan: nhãn là số lần trễ hạn của 11 khách có thu nhập, ba ứng viên là tuổi, thu nhập và dư nợ.

A filter by correlation: the label is the late-payment count of the 11 customers with income, and the three candidates are age, income and balance.

```
import math

def pearson(x, y):
    mx, my = sum(x) / len(x), sum(y) / len(y)
    tu = sum((a - mx) * (b - my) for a, b in zip(x, y))
    mau = math.sqrt(sum((a - mx) ** 2 for a in x) *
                    sum((b - my) ** 2 for b in y))
    return tu / mau

# 11 khách có thu nhập (bỏ KH07); nhãn: số lần trễ hạn
tre = [2, 1, 0, 1, 4, 0, 0, 1, 3, 0, 0]
cot = {"tuổi": [25, 29, 34, 41, 23, 38, 31, 27, 36, 50, 47],
       "thu nhập": [9, 12, 15, 22, 8, 18, 14, 11, 16, 20, 120],
       "dư nợ": [12, 18, 20, 45, 10, 30, 16, 14, 25, 35, 150]}
r = {c: pearson(v, tre) for c, v in cot.items()}
for c in sorted(r, key=lambda c: -abs(r[c])):
    print(f"{c:<10} r = {r[c]:.4f}" .replace(".", ","))
```

Kết quả chạy / Output

```
tuổi      r = -0,5671
dư nợ     r = -0,3520
thu nhập  r = -0,3309
```

Tuổi có $|r|$ lớn nhất (0,5671): khách càng lớn tuổi càng ít lần trễ hạn trong mẫu này. Dư nợ và thu nhập đứng sau với 0,3520 và 0,3309, và hai cột này tương quan 0,9898 với nhau (mục 3.5), nên giữ một trong hai là đủ. Mẫu 11 khách chỉ để minh họa cách làm, không đủ để kết luận.

Age has the largest $|r|$ (0.5671): the older the customer, the fewer late payments in this sample. Balance and income follow with 0.3520 and 0.3309, and these two correlate at 0.9898 with each other (section 3.5), so keeping one of them is enough. A sample of 11 customers only illustrates the method and is too small to conclude.

PCA (principal component analysis, phân tích thành phần chính) ghép các cột tương quan thành thành phần chính không tương quan. Thành phần thứ nhất (PC1) là hướng dọc theo đó dữ liệu phân tán nhiều nhất; thành phần thứ hai vuông góc với nó và giữ phần phân tán còn lại. Về mặt đại số, PC là véc-tơ riêng của ma trận hiệp phương sai, phần phân tán nó giữ là trị riêng tương ứng. Với hai thuộc tính, ma trận hiệp phương sai có dạng 2×2 và trị riêng tính được bằng công thức đóng, với a , d là phương sai hai cột và b là hiệp phương sai.

PCA (principal component analysis) combines correlated columns into uncorrelated principal components. The first component (PC1) is the direction along which the data spread most; the second is perpendicular to it and holds the remaining spread. Algebraically, a PC is an eigenvector of the covariance matrix, and the spread it holds is the matching eigenvalue. With two attributes, the covariance matrix is 2×2 and the eigenvalues have a closed formula, with a and d the variances of the two columns and b their covariance.

$$S = \begin{bmatrix} a & b \\ b & d \end{bmatrix}$$

$$\lambda = (a + d) / 2 \pm \sqrt{((a - d) / 2)^2 + b^2} \quad (3.8)$$

Véc-tơ riêng của $\lambda_1 \propto (b, \lambda_1 - a)$, chia cho độ dài; điểm trên PC1 = $e \cdot (x - \bar{x})$ (3.9)

Dữ kiện đề bài: 5 khách (P1 đến P5) có thu nhập $x = 8, 10, 12, 14, 16$ và dư nợ $y = 10, 12, 13, 17, 18$ (triệu đồng). Hai cột cùng đơn vị nên dùng ma trận hiệp phương sai; nếu khác đơn vị phải chuẩn hóa z-score trước, tương đương dùng ma trận tương quan.

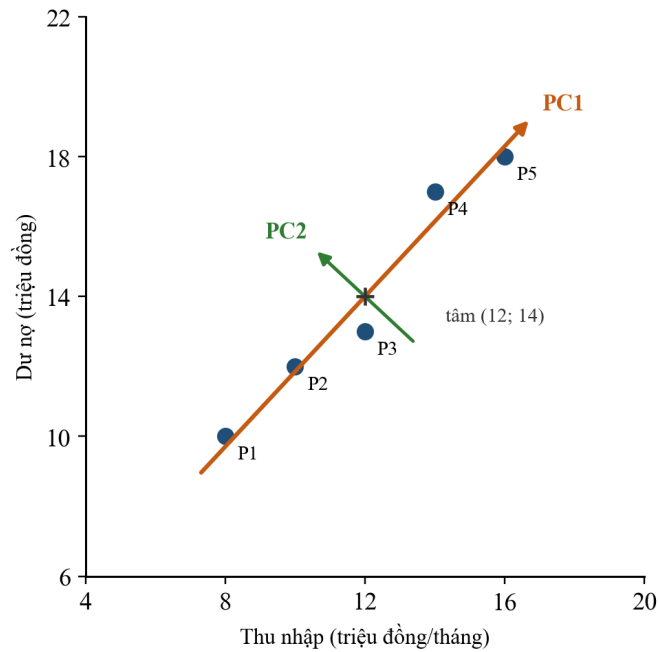
Given in the task: 5 customers (P1 to P5) with income $x = 8, 10, 12, 14, 16$ and balance $y = 10, 12, 13, 17, 18$ (million dong). The two columns share a unit, so the covariance matrix is used; with different units, z-score standardisation must come first, which amounts to using the correlation matrix.

```
import math

x = [8, 10, 12, 14, 16]      # thu nhập (triệu đồng/tháng)
y = [10, 12, 13, 17, 18]    # dư nợ (triệu đồng)
n = len(x)
mx, my = sum(x) / n, sum(y) / n
a = sum((u - mx) ** 2 for u in x) / (n - 1)
d = sum((v - my) ** 2 for v in y) / (n - 1)
b = sum((u - mx) * (v - my) for u, v in zip(x, y)) / (n - 1)
tam = (a + d) / 2
ban = math.sqrt(((a - d) / 2) ** 2 + b * b)
l1, l2 = tam + ban, tam - ban
print(f"Ma trận hiệp phương sai: [{a:.2f}, {b:.2f}], [{b:.2f}, {d:.2f}]]"
      .replace(".", ","))
print(f"Trị riêng: {l1:.4f} và {l2:.4f}".replace(".", ","))
print(f"PC1 giữ {l1 / (l1 + l2) * 100:.2f}% phương sai".replace(".", ","))
nm = math.hypot(b, l1 - a)
e1 = (b / nm, (l1 - a) / nm)
print(f"Véc-tơ riêng PC1: ({e1[0]:.4f}; {e1[1]:.4f})".replace(".", ","))
diem = [e1[0] * (u - mx) + e1[1] * (v - my) for u, v in zip(x, y)]
print("Điểm trên PC1:", [f"{t:.2f}".replace(".", ",") for t in diem])
```

Kết quả chạy / Output

```
Ma trận hiệp phương sai: [[10,00, 10,50], [10,50, 11,50]]
Trị riêng: 21,2768 và 0,2232
PC1 giữ 98,96% phương sai
Véc-tơ riêng PC1: (0,6815; 0,7319)
Điểm trên PC1: [-5,65', '-2,83', '-0,73', '3,56', '5,65']
```



Hình 3.4. Năm điểm thu nhập – dư nợ và hai thành phần chính
Figure 3.4. Five income–balance points and the two principal components

Bài mẫu 3.7. PCA cho 5 điểm — Worked example 3.7. PCA on 5 points

Bước Step	Việc làm Action	Phép tính Calculation	Kết quả Result
1	Trung bình <i>Means</i>	$\bar{x} = 60 / 5; \bar{y} = 70 / 5$	12; 14
2	Hiệp phương sai <i>Covariance</i>	$a = 40 / 4; d = 46 / 4; b = 42 / 4$ (mẫu số $n - 1 = 4$) $a = 40 / 4; d = 46 / 4; b = 42 / 4$ (denominator $n - 1 = 4$)	10; 11,5; 10,5
3	Trị riêng <i>Eigenvalues</i>	$(10 + 11,5) / 2 = 10,75; \sqrt{(0,75^2 + 10,5^2)} = \sqrt{110,8125} = 10,5268$	21,2768; 0,2232
4	Tỷ lệ PC1 <i>PC1 share</i>	$21,2768 / (21,2768 + 0,2232)$	98,96%
5	Véc-tơ PC1 <i>PC1 vector</i>	$(b, \lambda_1 - a) = (10,5; 11,2768)$, độ dài 15,4083 $(b, \lambda_1 - a) = (10,5; 11,2768)$, length 15.4083	(0,6815; 0,7319)
6	Điểm trên PC1 <i>Scores on PC1</i>	$0,6815 * (x - 12) + 0,7319 * (y - 14)$	-5,65; -2,83; -0,73; 3,56; 5,65

PC1 giữ 98,96% phương sai nên một cột điểm PC1 thay được hai cột gốc với mất mát nhỏ. Kiểm tra: tổng hai trị riêng $21,2768 + 0,2232 = 21,5$ bằng $a + d = 10 + 11,5$, vì tổng phương sai được bảo toàn.

PC1 holds 98.96% of the variance, so one column of PC1 scores can replace the two originals with little loss. Check: the sum of the eigenvalues $21.2768 + 0.2232 = 21.5$ equals $a + d = 10 + 11.5$, because total variance is conserved.

PHẢI THUỘC / MEMORISE — Tổng các trị riêng bằng tổng phương sai gốc (vết ma trận). Tỷ lệ phương sai giữ của PC = trị riêng của nó / tổng trị riêng. PCA nhạy với thang đo, nên chuẩn hóa trước khi áp dụng nếu các cột khác đơn vị.

The eigenvalues sum to the original total variance (the trace). The variance share of a PC = its eigenvalue / the sum of eigenvalues. PCA is sensitive to scale, so standardise first if the columns have different units.

3.9. Lấy mẫu và mất cân bằng lớp — 3.9. Sampling and class imbalance

Lấy mẫu (sampling) chọn một phần dữ liệu để thử nhanh hoặc để chia tập huấn luyện – kiểm tra. Lấy mẫu ngẫu nhiên đơn cho mọi dòng cùng xác suất được chọn. Lấy mẫu phân tầng (stratified) chia dữ liệu thành tầng theo một thuộc tính, như nhãn tốt hoặc xấu, rồi lấy cùng một tỷ lệ ở mỗi tầng. Khi lớp hiếm chỉ chiếm vài phần trăm, mẫu ngẫu nhiên đơn có thể không chứa dòng nào của lớp đó.

Sampling picks part of the data for quick trials or to split training and test sets. Simple random sampling gives every row the same chance of selection. Stratified sampling divides the data into strata by an attribute, such as a good or bad label, and takes the same share from each stratum. When the rare class is only a few percent, a simple random sample may hold no row of that class at all.

Dữ kiện đề bài: 100 hồ sơ vay, 90 hồ sơ tốt và 10 hồ sơ xấu (mã HS001 đến HS100; HS091 đến HS100 là hồ sơ xấu). Lấy mẫu 10% với random seed 2026 của Python 3.12.

Given in the task: 100 loan files, 90 good and 10 bad (codes HS001 to HS100; HS091 to HS100 are the bad files). A 10% sample with random seed 2026 in Python 3.12.

```
import random

nhan = ["tốt"] * 90 + ["xấu"] * 10
ma = [f"HS{i:03d}" for i in range(1, 101)]
rng = random.Random(2026)
don = rng.sample(range(100), 10)
xau = sum(1 for i in don if nhan[i] == "xấu")
print("Ngẫu nhiên đơn, 10 hồ sơ:", xau, "hồ sơ xấu")
tang = {"tốt": [i for i in range(100) if nhan[i] == "tốt"],
        "xấu": [i for i in range(100) if nhan[i] == "xấu"]}
phan_tang = {}
for t, ds in tang.items():
    phan_tang[t] = rng.sample(ds, round(len(ds) * 0.10))
print("Phân tầng 10%:", {t: len(v) for t, v in phan_tang.items()})
print("Mã hồ sơ xấu được chọn:", [ma[i] for i in phan_tang["xấu"]])
print("Giảm mẫu lớp tốt xuống 10:", 10 + 10, "hồ sơ")
print("Lấp lớp xấu lên 90:", 90 + 90, "hồ sơ")
a, b, lam = (9, 12), (8, 10), 0.4 # SMOTE: A, láng giềng B, hệ số
moi = [p + lam * (q - p) for p, q in zip(a, b)]
print("Điểm SMOTE:", f"({moi[0]:.1f}; {moi[1]:.1f}).replace(".", ",")")
```

Kết quả chạy / Output

```
Ngẫu nhiên đơn, 10 hồ sơ: 0 hồ sơ xấu
Phân tầng 10%: {'tốt': 9, 'xấu': 1}
Mã hồ sơ xấu được chọn: ['HS092']
Giảm mẫu lớp tốt xuống 10: 20 hồ sơ
Lặp lớp xấu lên 90: 180 hồ sơ
Điểm SMOTE: (8,6; 11,2)
```

Mẫu ngẫu nhiên đơn 10 hồ sơ không có hồ sơ xấu nào, nên mô hình học từ mẫu này không thấy lớp cần tìm. Mẫu phân tầng luôn có 9 hồ sơ tốt và 1 hồ sơ xấu (ở đây là HS092). Con số cụ thể phụ thuộc seed và thứ tự gọi hàm; đề thi phải nêu seed.

The simple random sample of 10 files holds no bad file, so a model learning from it never sees the class it must find. The stratified sample always has 9 good files and 1 bad file (here HS092). The exact figures depend on the seed and the order of calls; an exam must state the seed.

Mất cân bằng lớp (class imbalance) làm accuracy đánh lừa: dự đoán "tốt" cho cả 100 hồ sơ đúng 90%. Ba hướng xử lý: giảm mẫu lớp đa số (undersampling), tăng mẫu lớp thiểu số bằng cách lặp (oversampling), và SMOTE tạo dòng mới trên đoạn nối một điểm với láng giềng gần nhất cùng lớp. Gói imbalanced-learn có sẵn SMOTE; ở đây chỉ trình bày công thức.

Class imbalance makes accuracy misleading: predicting "good" for all 100 files is 90% right. Three remedies: reduce the majority class (undersampling), grow the minority class by repetition (oversampling), and SMOTE, which creates a new row on the segment joining a point to its nearest neighbour of the same class. The imbalanced-learn package provides SMOTE; here only the formula is shown.

$$x \text{ mới} = x + \lambda * (x \text{ láng giềng} - x), 0 \leq \lambda \leq 1 \quad (3.10)$$

Bài mẫu 3.8. Cân bằng lớp — *Worked example 3.8. Balancing the classes*

Đề bài: 90 hồ sơ tốt, 10 hồ sơ xấu. Hồ sơ xấu A có thu nhập 9, dư nợ 12; láng giềng gần nhất B có thu nhập 8, dư nợ 10; $\lambda = 0,4$. Tính số dòng sau cân bằng và điểm SMOTE.

Task: 90 good files, 10 bad. Bad file A has income 9 and balance 12; its nearest neighbour B has income 8 and balance 10; $\lambda = 0.4$. Find the row counts after balancing and the SMOTE point.

Bước Step	Cách Method	Phép tính Calculation	Kết quả Result
1	Giảm mẫu <i>Undersampling</i>	Giữ 10 hồ sơ tốt, 10 hồ sơ xấu <i>Keep 10 good, 10 bad</i>	20 dòng <i>20 rows</i>
2	Tăng mẫu lặp <i>Oversampling</i>	Lặp hồ sơ xấu đến 90 <i>Repeat bad files up to 90</i>	180 dòng <i>180 rows</i>
3	SMOTE, thu nhập <i>SMOTE, income</i>	$9 + 0,4 * (8 - 9)$	8,6
4	SMOTE, dư nợ <i>SMOTE, balance</i>	$12 + 0,4 * (10 - 12)$	11,2

LỖI HAY GẶP / COMMON MISTAKE — Tăng mẫu lớp thiểu số trước khi chia tập huấn luyện và kiểm tra. Bản sao của cùng một hồ sơ nằm ở cả hai tập, và độ chính xác trên tập kiểm tra cao một cách giả. Chia tập trước, chỉ cân bằng tập huấn luyện.

Oversampling the minority class before splitting into training and test sets. Copies of the same file land in both sets, and test accuracy looks falsely high. Split first and balance only the training set.

3.10. Rò rỉ dữ liệu và thứ tự đúng của các bước — 3.10. Data leakage and the correct order of steps

Rò rỉ dữ liệu (data leakage) xảy ra khi thông tin từ tập kiểm tra, hoặc từ tương lai, lọt vào bước huấn luyện. Mô hình có điểm cao khi thử nhưng thất bại khi chạy thật. Nguồn rò rỉ hay gặp ở tiền xử lý: tính trung bình, độ lệch chuẩn, min, max hoặc giá trị điền thiếu trên toàn bộ dữ liệu trước khi chia tập; đưa vào thuộc tính chỉ biết sau thời điểm cho vay; cùng một khách xuất hiện ở cả hai tập.

Data leakage occurs when information from the test set, or from the future, slips into training. The model scores well in trials but fails in production. Common sources in preprocessing: computing the mean, standard deviation, min, max or fill value on all the data before splitting; including an attribute known only after the loan date; the same customer appearing in both sets.

Quy trình đúng: chia tập trước; học tham số (trung bình, độ lệch chuẩn, min, max, giá trị điền) chỉ trên tập huấn luyện; áp tham số đó cho tập kiểm tra. Trong scikit-learn, hàm fit chỉ gọi trên tập huấn luyện, transform gọi trên cả hai. Ví dụ: tám khách đầu (KH01 đến KH09, trừ KH07) làm tập huấn luyện, ba khách còn lại (KH10 đến KH12) làm tập kiểm tra.

The correct procedure: split first; learn the parameters (mean, standard deviation, min, max, fill value) on the training set only; apply those parameters to the test set. In scikit-learn, fit is called on the training set only and transform on both. Example: the first eight customers (KH01 to KH09, excluding KH07) form the training set and the other three (KH10 to KH12) the test set.

```
import statistics as st

huan_luyen = [9, 12, 15, 22, 8, 18, 14, 11] # KH01-KH06, KH08, KH09
kiem_tra = [16, 20, 120] # KH10-KH12
toan_bo = huan_luyen + kiem_tra

def z(x, mau):
    return (x - st.mean(mau)) / st.stdev(mau)

def vn(x):
    return f"{x:.4f}".replace(".", ",")

print("z của 120, chuẩn hóa trên toàn bộ (rò rỉ):", vn(z(120, toan_bo)))
print("z của 120, chuẩn hóa trên tập huấn luyện:", vn(z(120, huan_luyen)))
lo, hi = min(huan_luyen), max(huan_luyen)
mm = [(x - lo) / (hi - lo) for x in kiem_tra]
print("Min-max tập kiểm tra:", [vn(v) for v in mm])
```

Kết quả chạy / Output

```
z của 120, chuẩn hóa trên toàn bộ (rò rỉ): 2,9868
z của 120, chuẩn hóa trên tập huấn luyện: 22,6884
Min-max tập kiểm tra: ['0,5714', '0,8571', '8,0000']
```

Khi chuẩn hóa trên toàn bộ, z của 120 là 2,9868: gần ngưỡng 3 và thoát khỏi quy tắc $|z| > 3$. Khi chuẩn hóa đúng trên tập huấn luyện, z của 120 là 22,6884: giá trị ở tập kiểm tra lệch hẳn khỏi những gì mô hình từng thấy. Giá trị lạ như vậy phải còn nguyên ở tập kiểm tra, vì dữ liệu thật sau này cũng sẽ có. Min–max theo huấn luyện đưa 16 và 20 về 0,5714 và 0,8571, còn 120 thành 8,0000, nằm ngoài $[0, 1]$; kết quả này đúng, và không được cắt lại chỉ để cho đẹp.

With normalisation on all the data, the z of 120 is 2.9868: close to the threshold of 3 and escaping the $|z| > 3$ rule. With correct normalisation on the training set, the z of 120 is 22.6884: the test value sits far from anything the model has seen. That strangeness has to survive in the test set, because real data will later contain such values too. Min–max by the training range takes 16 and 20 to 0.5714 and 0.8571, and 120 to 8.0000, outside $[0, 1]$; that outcome is correct and must not be clipped to look neat.

Bài mẫu 3.9. Chuẩn hóa không rò rỉ — *Worked example 3.9. Normalising without leakage*

Bước <i>Step</i>	Việc làm <i>Action</i>	Phép tính <i>Calculation</i>	Kết quả <i>Result</i>
1	Trung bình huấn luyện <i>Training mean</i>	$(9 + 12 + 15 + 22 + 8 + 18 + 14 + 11) / 8$ $= 109 / 8$	13,625
2	Tổng bình phương lệch <i>Sum of squared deviations</i>	$21,3906 + 2,6406 + 1,8906 + 70,1406 +$ $31,6406 + 19,1406 + 0,1406 + 6,8906$	153,875
3	Độ lệch chuẩn mẫu <i>Sample deviation</i>	$\sqrt{(153,875 / 7)} = \sqrt{21,9821}$	4,6885
4	z của 120 <i>z of 120</i>	$(120 - 13,625) / 4,6885$	22,6884
5	Min–max của 120 <i>Min–max of 120</i>	$\min = 8, \max = 22: (120 - 8) / (22 - 8)$	8

LỖI HAY GẶP / COMMON MISTAKE — Gọi fit trên toàn bộ bảng rồi mới chia tập huấn luyện – kiểm tra. Điểm kiểm tra cao, nhưng triển khai thật thì thấp vì tham số chuẩn hóa đã "nhìn thấy" dữ liệu kiểm tra.

Calling fit on the whole table and only then splitting into training and test sets. The test score is high, but real deployment scores low because the scaling parameters have already "seen" the test data.

PHẢI THUỘC / MEMORISE — Thứ tự: chia tập, rồi học tham số tiền xử lý trên tập huấn luyện, rồi áp cho tập kiểm tra. Lấy mẫu lại, SMOTE, chọn thuộc tính cũng chỉ làm sau khi chia tập, và chỉ trên tập huấn luyện.

Order: split, then learn the preprocessing parameters on the training set, then apply them to the test set. Resampling, SMOTE and feature selection also come after the split and only on the training set.

3.11. Bảng thuật ngữ — 3.11. Glossary

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Tiền xử lý dữ liệu	data preprocessing	Các thao tác đưa dữ liệu thô về dạng thuật toán dùng được <i>Operations that bring raw data into a form algorithms can use</i>
Giá trị thiếu	missing value	Ô không có dữ liệu <i>A cell that holds no data</i>
Điền khuyết	imputation	Thay ô thiếu bằng một giá trị ước lượng <i>Replacing a blank with an estimated value</i>
Nhiều	noise	Sai số ngẫu nhiên phủ lên giá trị đo <i>Random error laid over a measured value</i>
Ngoại lai	outlier	Giá trị lệch xa phần lớn dữ liệu <i>A value far from most of the data</i>
Chia giỏ	binning	Gom giá trị đã sắp thành các giỏ rồi làm trơn <i>Grouping sorted values into bins and then smoothing</i>
Giỏ đều độ rộng	equal-width bin	Giỏ có khoảng giá trị bằng nhau <i>A bin spanning an equal value range</i>
Giỏ đều tần số	equal-frequency bin	Giỏ có số phần tử gần bằng nhau <i>A bin holding a similar number of elements</i>
Khoảng tứ phân vị	interquartile range (IQR)	Hiệu $Q3 - Q1$ <i>The difference $Q3 - Q1$</i>
Điểm z	z-score	Số độ lệch chuẩn mà giá trị cách trung bình <i>How many standard deviations a value lies from the mean</i>
Tích hợp dữ liệu	data integration	Ghép dữ liệu từ nhiều nguồn thành một bảng <i>Merging data from several sources into one table</i>
Kiểm định chi-bình phương	chi-squared test	Kiểm tra hai thuộc tính định danh có độc lập không <i>Tests whether two nominal attributes are independent</i>
Chuẩn hóa min-max	min-max normalisation	Co giãn tuyến tính về một đoạn, thường $[0, 1]$ <i>Linear rescaling onto a range, usually $[0, 1]$</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Chuẩn hóa z-score	z-score normalisation	Trừ trung bình, chia độ lệch chuẩn <i>Subtract the mean, divide by the standard deviation</i>
Mã hóa one-hot	one-hot encoding	Mỗi giá trị định danh thành một cột 0/1 <i>Each nominal value becomes a 0/1 column</i>
Rời rạc hóa	discretisation	Đổi thuộc tính số thành các nhóm <i>Turning a numeric attribute into groups</i>
Xây dựng thuộc tính	feature construction	Tạo cột mới từ cột có sẵn <i>Creating a new column from existing ones</i>
Chọn thuộc tính	feature selection	Giữ tập con thuộc tính hữu ích <i>Keeping a useful subset of attributes</i>
Phân tích thành phần chính	principal component analysis (PCA)	Ghép cột tương quan thành thành phần không tương quan <i>Combining correlated columns into uncorrelated components</i>
Trị riêng	eigenvalue	Phương sai mà một thành phần chính giữ <i>The variance a principal component holds</i>
Lấy mẫu phân tầng	stratified sampling	Lấy cùng tỷ lệ ở mỗi tầng <i>Taking the same share from each stratum</i>
Mất cân bằng lớp	class imbalance	Một lớp chiếm tỷ lệ thấp <i>One class holds a low share</i>
Rò rỉ dữ liệu	data leakage	Thông tin kiểm tra hoặc tương lai lọt vào huấn luyện <i>Test or future information slipping into training</i>

3.12. Bài tập vận dụng có lời giải — *Applied exercises with solutions*

Bài 1. Chia giỏ dư nợ thẻ và làm trơn — *Exercise 1. Binning card balances and smoothing*

Dư nợ thẻ tín dụng của 9 khách (triệu đồng), theo thứ tự hồ sơ: 21, 4, 25, 8, 34, 15, 28, 24, 60. Chia thành 3 giỏ đều tần số (mỗi giỏ 3 giá trị), làm trơn theo trung bình, trung vị và biên giỏ. Sau đó chia 3 giỏ đều độ rộng và đếm số phần tử mỗi giỏ.

Credit-card balances of 9 customers (million dong), in file order: 21, 4, 25, 8, 34, 15, 28, 24, 60. Split into 3 equal-frequency bins (3 values each) and smooth by bin means, medians and boundaries. Then form 3 equal-width bins and count the elements of each.

- Sắp tăng dần: 4, 8, 15, 21, 24, 25, 28, 34, 60 ($n = 9$, ba giỏ, mỗi giỏ 3 giá trị).
Sort ascending: 4, 8, 15, 21, 24, 25, 28, 34, 60 ($n = 9$, three bins of 3 values).

2. Giỏ 1: 4, 8, 15; giỏ 2: 21, 24, 25; giỏ 3: 28, 34, 60.
Bin 1: 4, 8, 15; bin 2: 21, 24, 25; bin 3: 28, 34, 60.
3. Theo trung bình: $27 / 3 = 9$; $70 / 3 = 23,33$; $122 / 3 = 40,67$.
By means: $27 / 3 = 9$; $70 / 3 = 23.33$; $122 / 3 = 40.67$.
4. Theo trung vị: 8; 24; 34.
By medians: 8; 24; 34.
5. Theo biên: giỏ 1 thành 4, 4, 15 (8 cách 4 là 4, cách 15 là 7); giỏ 2 thành 21, 25, 25 (24 cách 25 là 1); giỏ 3 thành 28, 28, 60 (34 cách 28 là 6, cách 60 là 26).
By boundaries: bin 1 becomes 4, 4, 15 (8 is 4 from 4 and 7 from 15); bin 2 becomes 21, 25, 25 (24 is 1 from 25); bin 3 becomes 28, 28, 60 (34 is 6 from 28 and 26 from 60).
6. Đều độ rộng: $w = (60 - 4) / 3 = 18,67$; biên 4; 22,67; 41,33; 60. Đếm: 4, 8, 15, 21 vào giỏ 1; 24, 25, 28, 34 vào giỏ 2; 60 vào giỏ 3, tức 4, 4, 1.
Equal-width: $w = (60 - 4) / 3 = 18.67$; edges 4; 22.67; 41.33; 60. Count: 4, 8, 15, 21 in bin 1; 24, 25, 28, 34 in bin 2; 60 in bin 3, that is 4, 4, 1.

Bài 2. Ngoại lai trong thời gian giải ngân — Exercise 2. An outlier in disbursement time

Số giờ từ lúc nộp hồ sơ đến lúc giải ngân của 8 khoản vay: 6, 8, 9, 10, 12, 13, 15, 48. Tìm ngoại lai bằng IQR (quy ước inclusive) và bằng quy tắc $|z| > 3$ (độ lệch chuẩn mẫu). Sau khi bỏ ngoại lai, chuẩn hóa min-max giá trị 12.

Hours from filing to disbursement for 8 loans: 6, 8, 9, 10, 12, 13, 15, 48. Find outliers by the IQR (inclusive convention) and by the $|z| > 3$ rule (sample standard deviation). After removing the outlier, min-max normalise the value 12.

1. $n = 8$. Vị trí Q1: $(8 - 1) * 0,25 + 1 = 2,75$, giữa giá trị thứ 2 (8) và thứ 3 (9): $Q1 = 8,75$.
 $n = 8$. Position of Q1: $(8 - 1) * 0.25 + 1 = 2.75$, between the 2nd value (8) and the 3rd (9): $Q1 = 8.75$.
2. Vị trí Q3: $(8 - 1) * 0,75 + 1 = 6,25$, giữa giá trị thứ 6 (13) và thứ 7 (15): $Q3 = 13 + 0,25 * 2 = 13,5$.
Position of Q3: $(8 - 1) * 0.75 + 1 = 6.25$, between the 6th value (13) and the 7th (15): $Q3 = 13 + 0.25 * 2 = 13.5$.
3. $IQR = 13,5 - 8,75 = 4,75$; hàng rào $8,75 - 7,125 = 1,625$ và $13,5 + 7,125 = 20,625$. Giá trị 48 nằm ngoài: ngoại lai.
 $IQR = 13.5 - 8.75 = 4.75$; fences $8.75 - 7.125 = 1.625$ and $13.5 + 7.125 = 20.625$. The value 48 lies outside: an outlier.
4. Trung bình $121 / 8 = 15,125$; độ lệch chuẩn mẫu 13,5903; z của 48 = $(48 - 15,125) / 13,5903 = 2,4190$, nhỏ hơn 3 nên quy tắc z-score bỏ sót.
Mean $121 / 8 = 15.125$; sample standard deviation 13.5903; z of 48 = $(48 - 15.125) / 13.5903 = 2.4190$, less than 3, so the z-score rule misses it.
5. Bỏ 48, còn 7 giá trị: min = 6, max = 15. Min-max của 12 = $(12 - 6) / (15 - 6) = 6 / 9 = 0,6667$.
Without 48, 7 values remain: min = 6, max = 15. Min-max of 12 = $(12 - 6) / (15 - 6) = 6 / 9 = 0.6667$.

Bài 3. Hạng thẻ và mua bảo hiểm khoản vay — Exercise 3. Card tier and loan-insurance purchase

Trong 150 khách, nhóm hạng thẻ Vàng trở lên có 80 người, trong đó 44 người mua bảo hiểm khoản vay và 36 người không mua; nhóm hạng Chuẩn có 70 người, trong đó 22 mua và 48 không mua. Tính χ^2 và kết luận ở mức ý nghĩa 5% (ngưỡng 3,841, bậc tự do 1).

Among 150 customers, the Vàng-and-above group has 80 people, of whom 44 bought loan insurance and 36 did not; the Chuẩn group has 70 people, of whom 22 bought and 48 did not. Compute χ^2 and conclude at the 5% level (threshold 3.841, 1 degree of freedom).

- Tổng hàng: 80 và 70; tổng cột: mua 44 + 22 = 66, không mua 36 + 48 = 84; tổng chung 150.
Row totals: 80 and 70; column totals: bought 44 + 22 = 66, did not buy 36 + 48 = 84; grand total 150.
- Kỳ vọng: $80 * 66 / 150 = 35,2$; $80 * 84 / 150 = 44,8$; $70 * 66 / 150 = 30,8$; $70 * 84 / 150 = 39,2$.
Expected: $80 * 66 / 150 = 35.2$; $80 * 84 / 150 = 44.8$; $70 * 66 / 150 = 30.8$; $70 * 84 / 150 = 39.2$.
- Các số hạng $(O - E)^2 / E$: $(44 - 35,2)^2 / 35,2 = 2,2000$; $(36 - 44,8)^2 / 44,8 = 1,7286$; $(22 - 30,8)^2 / 30,8 = 2,5143$; $(48 - 39,2)^2 / 39,2 = 1,9755$.
The terms $(O - E)^2 / E$: $(44 - 35.2)^2 / 35.2 = 2.2000$; $(36 - 44.8)^2 / 44.8 = 1.7286$; $(22 - 30.8)^2 / 30.8 = 2.5143$; $(48 - 39.2)^2 / 39.2 = 1.9755$.
- $\chi^2 = 8,4184$ lớn hơn 3,841 nên bác bỏ độc lập: hạng thẻ và việc mua bảo hiểm có liên hệ. Tỷ lệ mua: nhóm Vàng trở lên $44 / 80 = 55\%$, nhóm Chuẩn $22 / 70 = 31,43\%$.
 $\chi^2 = 8.4184$ exceeds 3.841, so independence is rejected: card tier and insurance purchase are related. Purchase rate: Vàng and above $44 / 80 = 55\%$, Chuẩn $22 / 70 = 31.43\%$.

3.13. Câu hỏi tự luận và dàn ý trả lời — Essay questions and answer outlines

Câu 1. Phân biệt nhiễu và ngoại lai trong tiền xử lý dữ liệu. Vì sao không nên xóa mọi ngoại lai một cách tự động? Lấy ví dụ từ dữ liệu khách hàng của Phú Sa.

Question 1. Distinguish noise from outliers in data preprocessing. Why should outliers not be deleted automatically? Give an example from the Phú Sa customer data.

- Nhiễu là sai số ngẫu nhiên nhỏ phủ lên nhiều giá trị; ngoại lai là vài giá trị lệch xa.
Noise is small random error over many values; outliers are a few distant values.
- Xử lý nhiễu: chia giỏ và làm trơn, trung bình trượt.
Treating noise: binning and smoothing, moving averages.
- Xử lý ngoại lai: kiểm nguồn, giữ có đánh dấu, cắt đầu mút, biến đổi, hoặc mô hình riêng.
Treating outliers: check the source, keep with a flag, winsorise, transform or model separately.
- Ngoại lai có thể là thông tin thật: gian lận, khách giàu bất thường.
An outlier can be real information: fraud, an unusually rich customer.

- Ví dụ KH12: thu nhập 120, dư nợ 150, DTI 1,25 giống các khách khác, nên là trường hợp thật cần giữ.
Example KH12: income 120, balance 150, DTI 1.25 like the other customers, so a genuine case worth keeping.
- Xóa tự động làm mất đúng nhóm khách giá trị cao mà mô hình cần phục vụ.
Automatic deletion loses exactly the high-value customers the model must serve.

Câu 2. So sánh chuẩn hóa min–max và z-score: công thức, kết quả, độ nhạy với ngoại lai, trường hợp nên chọn.

Question 2. Compare min–max and z-score normalisation: formula, output, sensitivity to outliers, and when to choose each.

- Min–max: $(x - \min) / (\max - \min)$, kết quả trong $[0, 1]$.
Min–max: $(x - \min) / (\max - \min)$, output within $[0, 1]$.
- Z-score: $(x - \text{trung bình}) / \text{độ lệch chuẩn mẫu}$, trung bình 0 và độ lệch chuẩn 1, không bị chặn.
Z-score: $(x - \text{mean}) / \text{sample standard deviation}$, mean 0 and standard deviation 1, unbounded.
- Ngoại lai: min–max bị min, max chi phối; với thu nhập Phú Sa, mười khách bị nén vào $[0; 0,125]$.
Outliers: min–max is governed by min and max; with Phú Sa income, ten customers are squeezed into $[0; 0.125]$.
- Chọn min–max khi cần thang cố định, z-score khi dữ liệu gần chuông hoặc thuật toán giả định chuẩn.
Choose min–max for a fixed scale, z-score when the data are near bell-shaped or the algorithm assumes normality.
- Cả hai không xóa ngoại lai và cần fit trên tập huấn luyện.
Neither removes outliers and both must be fitted on the training set.

Câu 3. Trình bày rò rỉ dữ liệu trong tiền xử lý: định nghĩa, ba nguồn thường gặp, hậu quả và quy trình đúng.

Question 3. Describe data leakage in preprocessing: definition, three common sources, consequences and the correct procedure.

- Định nghĩa: thông tin từ tập kiểm tra hoặc tương lai lọt vào huấn luyện.
Definition: information from the test set or the future slips into training.
- Nguồn 1: tính trung bình, độ lệch chuẩn, min, max, giá trị điền trên toàn bộ dữ liệu trước khi chia tập.
Source 1: computing the mean, standard deviation, min, max or fill value on all data before splitting.
- Nguồn 2: thuộc tính chỉ biết sau thời điểm cho vay. Nguồn 3: cùng một khách ở cả hai tập, hoặc cân bằng lớp trước khi chia.
Source 2: an attribute known only after the loan date. Source 3: the same customer in both sets, or balancing classes before the split.

- Hậu quả: điểm kiểm tra cao giả, mô hình kém khi chạy thật.
Consequence: falsely high test score, a weak model in production.
- Quy trình đúng: chia tập, fit trên huấn luyện, transform cho kiểm tra; ví dụ z của 120 là 22,6884 thay vì 2,9868.
Correct procedure: split, fit on training, transform for test; for example the z of 120 is 22.6884 rather than 2.9868.

3.14. Câu hỏi trắc nghiệm — Multiple-choice questions

17. Hạng thẻ được ghi "Vàng" ở hệ thống thẻ nhưng "Gold" ở ứng dụng. Lỗi này thuộc tiêu chí chất lượng dữ liệu nào?
A card tier is recorded as "Vàng" in the card system but "Gold" in the app. Which data-quality dimension does this fault belong to?
- A. Chính xác / *Accuracy*
 - B. Đầy đủ / *Completeness*
 - C. Nhất quán / *Consistency*
 - D. Kịp thời / *Timeliness*
18. Thu nhập 11 khách có số liệu, sắp tăng dần (triệu đồng): 8, 9, 11, 12, 14, 15, 16, 18, 20, 22, 120. Điền thu nhập thiếu của KH07 bằng trung vị thì được giá trị nào?
The incomes of the 11 customers with data, sorted (million dong): 8, 9, 11, 12, 14, 15, 16, 18, 20, 22, 120. Which value results from filling the missing income of KH07 with the median?
- A. 24,09 / *24.09*
 - B. 22
 - C. 18
 - D. 15
19. Thu nhập 22 triệu đồng của KH04, trung bình mẫu 24,09 và độ lệch chuẩn mẫu 32,11. Điểm z của KH04 là bao nhiêu?
KH04 has an income of 22 million dong, the sample mean is 24.09 and the sample standard deviation 32.11. What is the z-score of KH04?
- A. 0,125 / *0.125*
 - B. 0,0651 / *0.0651*
 - C. -0,0651 / *-0.0651*
 - D. 2,9868 / *2.9868*
20. Theo quy tắc IQR, giá trị x được coi là ngoại lai khi nào?
Under the IQR rule, when is a value x treated as an outlier?
- A. $x > Q3 + IQR$
 - B. x ngoài khoảng $[Q1 - IQR, Q3 + IQR]$ / *x outside $[Q1 - IQR, Q3 + IQR]$*
 - C. $|z| > 1,5$ / *$|z| > 1.5$*
 - D. $x < Q1 - 1,5 * IQR$ hoặc $x > Q3 + 1,5 * IQR$ / *$x < Q1 - 1.5 * IQR$ or $x > Q3 + 1.5 * IQR$*

- 21.** Dư nợ thẻ của 9 khách có min = 4, max = 60. Chuẩn hóa min–max giá trị 21 cho kết quả nào (làm tròn 4 chữ số)?
Card balances of 9 customers have min = 4 and max = 60. What does min–max normalisation of the value 21 give (rounded to 4 digits)?
- A. 0,3750 / 0.3750
 - B. 0,3036 / 0.3036
 - C. 0,2833 / 0.2833
 - D. 0,5000 / 0.5000
- 22.** Một giỏ gồm 4, 8, 15. Làm tròn theo biên giỏ cho kết quả nào?
A bin holds 4, 8, 15. What does smoothing by bin boundaries give?
- A. 9, 9, 9
 - B. 4, 4, 15
 - C. 8, 8, 8
 - D. 4, 15, 15
- 23.** Ma trận hiệp phương sai của hai thuộc tính có trị riêng 6 và 2. Thành phần chính thứ nhất giữ bao nhiêu phần trăm phương sai?
The covariance matrix of two attributes has eigenvalues 6 and 2. What percentage of the variance does the first principal component hold?
- A. 25%
 - B. 33,3% / 33.3%
 - C. 75%
 - D. 50%
- 24.** Cách chuẩn hóa z-score nào tránh rò rỉ dữ liệu khi có tập huấn luyện và tập kiểm tra?
Which way of z-score normalisation avoids data leakage when there are training and test sets?
- A. Tính trung bình, độ lệch chuẩn trên tập huấn luyện rồi áp cho tập kiểm tra / *Compute the mean and deviation on the training set and apply them to the test set*
 - B. Tính trên toàn bộ dữ liệu rồi mới chia hai tập / *Compute on all the data and only then split the two sets*
 - C. Tính riêng trên tập kiểm tra rồi áp cho chính nó / *Compute on the test set alone and apply to itself*
 - D. Tính trên tập kiểm tra rồi áp cho tập huấn luyện / *Compute on the test set and apply to the training set*

Chương 4. Khai phá luật kết hợp — Chapter 4. Association rule mining

Mục tiêu chương — Chapter objectives

- Phân biệt tập mục, độ hỗ trợ, độ tin cậy, lift, leverage và conviction; tính từng độ đo trên 10 giao dịch sản phẩm của Phú Sa.
Tell apart itemset, support, confidence, lift, leverage and conviction, and compute each on the 10 Phú Sa product transactions.
- Chạy tay Apriori từ C1 đến L3 với minsup 30%, giải thích bước ghép và tỉa bằng tính chất đóng xuống.
Run Apriori by hand from C1 to L3 with minsup 30% and explain the join and prune steps through downward closure.
- Sinh luật từ tập phổ biến, lọc theo minconf 70% và loại luật mạnh nhưng có lift thấp.
Generate rules from frequent itemsets, filter by minconf 70% and discard strong rules with low lift.
- Phân biệt tập phổ biến đóng và cực đại; dựng FP-tree và khai phá bằng cơ sở mẫu điều kiện.
Distinguish closed from maximal frequent itemsets; build an FP-tree and mine it with conditional pattern bases.
- So sánh Apriori với FP-growth; nêu luật nhiều mức, mẫu tuần tự, ứng dụng đề xuất sản phẩm và giới hạn của luật kết hợp.
Compare Apriori with FP-growth; state multilevel rules, sequential patterns, product recommendation and the limits of association rules.

4.1. Bài toán phân tích giỏ hàng và bán chéo sản phẩm tài chính — 4.1. Market basket analysis and cross-selling of financial products

Khai phá luật kết hợp (association rule mining) tìm các quy luật dạng "giao dịch có sản phẩm X thì thường có thêm sản phẩm Y" trong một tập giao dịch. Bài toán không có nhãn lớp để dự đoán, nên thuộc nhóm học không giám sát và nhiệm vụ mô tả. Cái tên gốc là phân tích giỏ hàng (market basket analysis): mỗi lần thanh toán ở siêu thị là một giỏ, mỗi mặt hàng là một mục, và người quản lý muốn biết các mặt hàng nào hay nằm cùng giỏ để bố trí kệ, làm khuyến mại gộp.

Association rule mining looks for rules of the form "transactions containing product X tend to contain product Y as well" in a set of transactions. There is no class label to predict, so the task is unsupervised and descriptive. The original name is market basket analysis: each supermarket checkout is a basket, each article is an item, and the manager wants to know which articles often sit in the same basket in order to arrange shelves and build bundled offers.

Ở Công ty Tài chính TNHH Phú Sa, giỏ hàng là tập sản phẩm tài chính mà một khách đăng ký trong cùng một lần giao dịch, và mục tiêu là bán chéo (cross-selling): đề xuất đúng sản phẩm thứ hai cho khách đã có sản phẩm thứ nhất. Năm sản phẩm được mã hóa như bảng sau; bộ dữ liệu của chương gồm 10 giao dịch, ký hiệu GD1 đến GD10, và là dữ kiện đề bài dùng xuyên suốt các bài mẫu.

At Phú Sa Finance Company Ltd, a basket is the set of financial products one customer signs up for in a single transaction, and the goal is cross-selling: offering the right second product to a customer who already holds the first. The five products are coded as in the next table; the data set of the chapter has 10 transactions, GD1 to GD10, and is the exercise data used by every worked example.

Mã Code	Sản phẩm Product	Số giao dịch chứa Transactions containing it	Tỷ lệ Share
BH	bảo hiểm khoản vay <i>loan insurance</i>	8	80%
TD	thẻ tín dụng <i>credit card</i>	6	60%
VM	vay tiền mặt <i>cash loan</i>	6	60%
DM	trả góp điện máy <i>electronics instalment</i>	3	30%
XM	trả góp xe máy <i>motorbike instalment</i>	3	30%

Có hai cách viết cùng một dữ liệu. Dạng ngang liệt kê mỗi giao dịch là một tập mục, như $GD1 = \{TD, VM, BH\}$. Dạng ma trận 0 và 1 đặt giao dịch ở hàng, sản phẩm ở cột, và ghi 1 nếu giao dịch có sản phẩm đó (Hình 4.1). Trong cơ sở dữ liệu quan hệ, dạng thứ ba là bảng dọc: mỗi dòng là một cặp (mã giao dịch, mã sản phẩm). Khi dữ liệu nằm ở bảng dọc, số giao dịch chứa một sản phẩm là số mã giao dịch khác nhau, nên câu truy vấn phải dùng COUNT(DISTINCT).

The same data can be written in two ways. The horizontal form lists each transaction as an itemset, such as $GD1 = \{TD, VM, BH\}$. The 0-1 matrix puts transactions in rows and products in columns and writes 1 if the transaction holds the product (Figure 4.1). In a relational database a third form is the vertical table: each row is a pair (transaction code, product code). When the data sit in a vertical table, the number of transactions containing a product is the number of distinct transaction codes, so the query must use COUNT(DISTINCT).

```
SELECT ma_sp, COUNT(DISTINCT ma_gd) AS so_gd
FROM giao_dich_sp
GROUP BY ma_sp
ORDER BY so_gd DESC;
```

	TD (6)	VM (6)	DM (3)	XM (3)	BH (8)
GD1	1	1	0	0	1
GD2	1	0	0	0	1
GD3	0	1	1	0	1
GD4	1	1	0	0	1
GD5	0	0	0	1	1
GD6	1	0	1	0	0
GD7	0	1	0	1	1
GD8	1	1	0	0	0
GD9	0	0	1	1	1
GD10	1	1	0	0	1

Số trong ngoặc: tổng số giao dịch chứa sản phẩm đó

Hình 4.1. Ma trận 0 và 1 của 10 giao dịch và 5 sản phẩm; tên cột kèm số giao dịch chứa sản phẩm

Figure 4.1. The 0-1 matrix of 10 transactions and 5 products; each column name carries the number of transactions containing the product

Mã sau nhập 10 giao dịch bằng tập (set) của Python, đếm số giao dịch chứa từng sản phẩm và tính cỡ giỏ trung bình.

The code that follows enters the 10 transactions as Python sets, counts the transactions containing each product and computes the average basket size.

```
GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
TEN = {"TD": "thẻ tín dụng", "VM": "vay tiền mặt",
       "DM": "trả góp điện máy", "XM": "trả góp xe máy",
       "BH": "bảo hiểm khoản vay"}
n = len(GD)
tan = {ma: sum(1 for t in GD if ma in t) for ma in TEN}
print("Số giao dịch:", n)
for ma, c in sorted(tan.items(), key=lambda x: -x[1]):
    print(f"{ma} {TEN[ma]:<20} {c:>2} giao dịch {c / n:.0%}")
print("Cỡ giỏ trung bình:", sum(len(t) for t in GD) / n)
```

Kết quả chạy / Output

```
Số giao dịch: 10
BH  bảo hiểm khoản vay      8 giao dịch  80%
TD  thẻ tín dụng            6 giao dịch  60%
VM  vay tiền mặt            6 giao dịch  60%
DM  trả góp điện máy        3 giao dịch  30%
XM  trả góp xe máy          3 giao dịch  30%
Cỡ giỏ trung bình: 2.6
```

Bảo hiểm khoản vay có mặt trong 8 giao dịch (80%), thẻ tín dụng và vay tiền mặt mỗi sản phẩm 6 giao dịch (60%), hai sản phẩm trả góp mỗi sản phẩm 3 giao dịch (30%). Mỗi giỏ có trung bình 2,6 sản phẩm. Các con số này là tần số riêng lẻ; luật kết hợp quan tâm đến tần số của các sản phẩm xuất hiện cùng nhau.

Loan insurance appears in 8 transactions (80%), the credit card and the cash loan in 6 each (60%), and each of the two instalment products in 3 (30%). A basket holds 2.6 products on average. These are single-product frequencies; association rules care about the frequency of products that appear together.

LỖI HAY GẶP / COMMON MISTAKE — Đếm số dòng của bảng dọc thay vì số mã giao dịch khác nhau. Nếu một giao dịch ghi sản phẩm hai lần, hoặc bảng nối với bảng khác làm nhân dòng, độ hỗ trợ bị phóng to và có thể vượt 100%; luôn đếm DISTINCT theo mã giao dịch rồi chia cho tổng số giao dịch.

Counting the rows of a vertical table instead of the distinct transaction codes. If a transaction lists a product twice, or a join multiplies rows, support is inflated and can exceed 100%; always count DISTINCT transaction codes and divide by the total number of transactions.

4.2. Tập mục, độ hỗ trợ, độ tin cậy, lift và các độ đo khác — 4.2. Itemsets, support, confidence, lift and other measures

Gọi $I = \{TD, VM, DM, XM, BH\}$ là tập các mục (item), D là cơ sở dữ liệu gồm n giao dịch, mỗi giao dịch T là một tập con của I . Một tập gồm k mục gọi là k -tập mục (k -itemset). Số giao dịch của D chứa tập X gọi là số đếm hỗ trợ (support count) của X ; chia cho n ta có độ hỗ trợ (support). Tập X là tập phổ biến (frequent itemset) nếu độ hỗ trợ của nó không nhỏ hơn ngưỡng hỗ trợ tối thiểu minsup.

Let $I = \{TD, VM, DM, XM, BH\}$ be the set of items and D a database of n transactions, each transaction T being a subset of I . A set of k items is a k -itemset. The number of transactions in D that contain X is the support count of X ; dividing by n gives the support. X is a frequent itemset if its support is at least the minimum support threshold minsup.

$$\text{support}(X) = |\{T \in D : X \subseteq T\}| / n \quad (4.1)$$

Luật kết hợp có dạng $X \Rightarrow Y$, trong đó X và Y là hai tập mục khác rỗng không giao nhau; X là vế trái (tiền đề), Y là vế phải (hệ quả). Hai độ đo đầu tiên của luật là độ hỗ trợ và độ tin cậy (confidence).

An association rule has the form $X \Rightarrow Y$, where X and Y are non-empty disjoint itemsets; X is the antecedent and Y the consequent. The first two measures of a rule are support and confidence.

$$\text{support}(X \Rightarrow Y) = \text{support}(X \cup Y) \quad (4.2)$$

$$\text{confidence}(X \Rightarrow Y) = \text{support}(X \cup Y) / \text{support}(X) \quad (4.3)$$

Độ hỗ trợ của luật cho biết luật áp dụng được cho bao nhiêu phần giao dịch. Độ tin cậy là xác suất có điều kiện: trong các giao dịch có X , bao nhiêu phần có thêm Y . Một luật là luật mạnh (strong rule) nếu đạt cả minsup và ngưỡng tin cậy tối thiểu minconf. Với Phú Sa, minsup = 30% (tức 3 trong 10 giao dịch) và minconf = 70%.

The support of a rule tells for what share of transactions the rule applies. Confidence is a conditional probability: among the transactions with X, what share also contain Y. A rule is strong if it meets both minsup and the minimum confidence threshold minconf. For Phú Sa, minsup = 30% (3 of 10 transactions) and minconf = 70%.

Một luật có độ tin cậy cao vẫn có thể vô ích. Nếu 80% giao dịch vốn đã có bảo hiểm khoản vay, thì luật "vay tiền mặt $\Rightarrow$ bảo hiểm" với độ tin cậy 83,33% chỉ hơn mức nền 80% một chút. Độ nâng (lift) so độ tin cậy với tần suất nền của vế phải.

A rule with high confidence can still be useless. If 80% of transactions already carry loan insurance, the rule "cash loan $\Rightarrow$ insurance" with confidence 83.33% beats the 80% base rate by a small margin. Lift compares confidence with the base frequency of the consequent.

$$\text{lift}(X \Rightarrow Y) = \text{confidence}(X \Rightarrow Y) / \text{support}(Y) = \text{support}(X \cup Y) / (\text{support}(X) * \text{support}(Y)) \quad (4.4)$$

Giá trị lift <i>Lift value</i>	Ý nghĩa <i>Meaning</i>	Ví dụ trong bộ dữ liệu <i>Example in the data set</i>
lift > 1	X và Y xuất hiện cùng nhau nhiều hơn mức kỳ vọng nếu độc lập: tương quan dương <i>X and Y occur together more often than expected under independence: positive correlation</i>	XM $\Rightarrow$ BH: 1,25
lift = 1	X và Y độc lập; luật không cung cấp thông tin thêm <i>X and Y are independent; the rule adds no information</i>	TD $\Rightarrow$ VM, BH: 1,00
lift < 1	X và Y xuất hiện cùng nhau ít hơn mức kỳ vọng: tương quan âm <i>X and Y occur together less often than expected: negative correlation</i>	TD, VM $\Rightarrow$ BH: 0,9375

Hai độ đo bổ sung cùng khai thác sự lệch khỏi tính độc lập. Leverage lấy hiệu giữa tần suất xuất hiện chung thực tế và tần suất kỳ vọng nếu độc lập; leverage bằng 0 là độc lập. Conviction đo mức luật sai ít hơn so với khi X và Y độc lập; conviction bằng 1 là độc lập, và tiến tới vô cực khi độ tin cậy bằng 1.

Two further measures also exploit the departure from independence. Leverage is the difference between the observed joint frequency and the frequency expected under independence; zero leverage means independence. Conviction measures how much less often the rule fails compared with the case where X and Y are independent; a conviction of 1 means independence, and it tends to infinity when confidence is 1.

$$\text{leverage}(X \Rightarrow Y) = \text{support}(X \cup Y) - \text{support}(X) * \text{support}(Y) \quad (4.5)$$

$$\text{conviction}(X \Rightarrow Y) = (1 - \text{support}(Y)) / (1 - \text{confidence}(X \Rightarrow Y)) \quad (4.6)$$

Mã sau cài đặt năm độ đo cho bốn luật: vay tiền mặt $\Rightarrow$ bảo hiểm và chiều ngược lại, thẻ tín dụng + vay tiền mặt $\Rightarrow$ bảo hiểm, trả góp xe máy $\Rightarrow$ bảo hiểm. Python in vô cực là inf.

The code that follows implements the five measures for four rules: cash loan $\Rightarrow$ insurance and its reverse, credit card + cash loan $\Rightarrow$ insurance, and motorbike instalment $\Rightarrow$ insurance. Python prints infinity as inf.

```

GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
n = len(GD)

def dem(tap):
    return sum(1 for t in GD if tap <= t)

def do_luat(a, b):
    n_ab, n_a, n_b = dem(a | b), dem(a), dem(b)
    sup = n_ab / n
    conf = n_ab / n_a
    lift = conf / (n_b / n)
    lev = sup - (n_a / n) * (n_b / n)
    conv = (1 - n_b / n) / (1 - conf) if conf < 1 else float("inf")
    return sup, conf, lift, lev, conv

for a, b in [({"VM"}, {"BH"}), ({"BH"}, {"VM"}),
             ({"TD", "VM"}, {"BH"}), ({"XM"}, {"BH"})]:
    s, c, l, v, cv = do_luat(a, b)
    print(" ".join(sorted(a)), "=>", " ".join(sorted(b)))
    print(f" sup={s:.2f} conf={c:.4f} lift={l:.4f}"
          f" lev={v:+.2f} conv={cv:.4f}")

```

Kết quả chạy / Output

```

VM => BH
  sup=0.50 conf=0.8333 lift=1.0417 lev=+0.02 conv=1.2000
BH => VM
  sup=0.50 conf=0.6250 lift=1.0417 lev=+0.02 conv=1.0667
TD VM => BH
  sup=0.30 conf=0.7500 lift=0.9375 lev=-0.02 conv=0.8000
XM => BH
  sup=0.30 conf=1.0000 lift=1.2500 lev=+0.06 conv=inf

```

Luật $VM \Rightarrow BH$ có độ hỗ trợ 0,5, độ tin cậy 0,8333 (5 trên 6 giao dịch có VM), lift 1,0417, leverage 0,02 và conviction 1,2. Chiều ngược lại $BH \Rightarrow VM$ có cùng độ hỗ trợ và cùng lift nhưng độ tin cậy chỉ 0,625 (5 trên 8 giao dịch có BH). Luật $TD, VM \Rightarrow BH$ có độ tin cậy 0,75 nhưng lift 0,9375, leverage -0,02 và conviction 0,8: khách có cả thẻ và vay tiền mặt ít mua bảo hiểm hơn mức nền. Luật $XM \Rightarrow BH$ có độ tin cậy 1 nên conviction không xác định (vô cực).

Rule $VM \Rightarrow BH$ has support 0.5, confidence 0.8333 (5 of the 6 transactions with VM), lift 1.0417, leverage 0.02 and conviction 1.2. The reverse $BH \Rightarrow VM$ has the same support and lift but confidence of only 0.625 (5 of the 8 transactions with BH). Rule $TD, VM \Rightarrow BH$ has confidence 0.75 but lift 0.9375, leverage -0.02 and conviction 0.8: customers with both the card and the cash loan buy insurance less often than the base rate. Rule $XM \Rightarrow BH$ has confidence 1, so its conviction is undefined (infinite).

Độ đo <i>Measure</i>	Trả lời câu hỏi <i>Question answered</i>	Điểm yếu <i>Weakness</i>
Support	Luật áp dụng cho bao nhiêu phần giao dịch? <i>For what share of transactions does the rule apply?</i>	Bỏ sót luật hiếm nhưng giá trị cao <i>Misses rare rules of high value</i>
Confidence	Có X thì chắc chắn đến đâu có Y? <i>Given X, how likely is Y?</i>	Bỏ qua tần suất nền của Y; không đối xứng <i>Ignores the base frequency of Y; not symmetric</i>
Lift	Y xuất hiện cùng X nhiều hơn mức nền bao nhiêu lần? <i>How many times more often does Y occur with X than at the base rate?</i>	Đối xứng, nhạy với tập hiếm <i>Symmetric, sensitive to rare sets</i>

PHẢI THUỘC / MEMORISE — Support đo độ phổ biến, confidence đo độ chắc chắn theo chiều từ X sang Y, lift đo độ đáng quan tâm so với mức nền. Luật mạnh chỉ cần support và confidence; muốn biết luật có thú vị hay không phải xét lift lớn hơn 1.

Support measures how common, confidence measures how certain from X to Y, lift measures how interesting against the base rate. A strong rule needs only support and confidence; to know whether it is interesting, check that lift is above 1.

LỖI HAY GẶP / COMMON MISTAKE — Cho rằng độ tin cậy có tính đối xứng. Trong dữ liệu Phú Sa, VM $\Rightarrow$ BH có độ tin cậy $5/6 = 83,33\%$ còn BH $\Rightarrow$ VM chỉ $5/8 = 62,5\%$; mẫu số là số giao dịch có về TRÁI, nên đổi chiều luật là đổi mẫu số. Lift và support thì không đổi khi đổi chiều.

Assuming that confidence is symmetric. In the Phú Sa data VM $\Rightarrow$ BH has confidence $5/6 = 83.33\%$ while BH $\Rightarrow$ VM has only $5/8 = 62.5\%$; the denominator is the number of transactions holding the LEFT side, so reversing a rule changes the denominator. Lift and support do not change when the rule is reversed.

4.3. Thuật toán Apriori: tính chất đóng xuống, ghép và tỉa — 4.3. The Apriori algorithm: downward closure, join and prune

Với m mục, số tập mục khác rỗng được tính theo công thức (4.7). Năm sản phẩm của Phú Sa cho 31 tập mục; một cửa hàng 100 mặt hàng cho một số có 31 chữ số. Đếm hết là không thể, nên cần một tính chất cho phép loại sớm phần lớn tập mục mà không cần đếm.

With m items the number of non-empty itemsets follows formula (4.7). The five Phú Sa products give 31 itemsets; a shop with 100 articles gives a number with 31 digits. Counting them all is impossible, so a property is needed that discards most itemsets early without counting.

$$\text{số tập mục khác rỗng} = 2^m - 1 \quad (4.7)$$

Tính chất Apriori (còn gọi là tính chất đóng xuống, downward closure, hay tính đơn điệu giảm, anti-monotone): mọi tập con của một tập phổ biến cũng là tập phổ biến. Nói theo chiều ngược lại: nếu một tập mục không phổ biến thì mọi tập chứa nó cũng không phổ

biến. Lý do: giao dịch chứa tập lớn thì chứa mọi tập con của nó, nên số đếm của tập con không bao giờ nhỏ hơn số đếm của tập lớn.

The Apriori property (also called downward closure or anti-monotonicity): every subset of a frequent itemset is also frequent. Stated the other way round: if an itemset is not frequent, no itemset containing it is frequent. The reason is that a transaction containing the larger set contains every subset of it, so the count of a subset is never smaller than the count of the larger set.

$$X \subseteq Y \Rightarrow \text{support}(X) \geq \text{support}(Y) \quad (4.8)$$

Apriori đi theo từng mức $k = 1, 2, 3, \dots$, mỗi mức gồm bốn việc: (1) sinh tập ứng viên C_k từ các tập phổ biến L_{k-1} của mức trước; (2) tĩa ứng viên có tập con cỡ $k-1$ không thuộc L_{k-1} ; (3) quét cơ sở dữ liệu một lần để đếm các ứng viên còn lại; (4) giữ ứng viên có số đếm không nhỏ hơn ngưỡng làm L_k . Thuật toán dừng khi L_k rỗng hoặc không sinh được ứng viên.

Apriori proceeds level by level, $k = 1, 2, 3, \dots$, and each level has four tasks: (1) generate the candidate set C_k from the frequent sets L_{k-1} of the previous level; (2) prune any candidate with a $(k-1)$ -subset that is not in L_{k-1} ; (3) scan the database once to count the remaining candidates; (4) keep as L_k the candidates whose count is at least the threshold. The algorithm stops when L_k is empty or no candidate can be generated.

Bước ghép (join) lấy hai tập trong L_{k-1} có $k-2$ mục chung và hợp chúng lại thành tập cỡ k . Các bản cài đặt thường sắp mục theo thứ tự cố định và chỉ ghép hai tập trùng $k-2$ mục đầu; cách này sinh mỗi ứng viên đúng một lần, và sau bước tĩa cho cùng tập C_k với cách ghép mọi cặp. Bước tĩa (prune) là chỗ dùng tính chất đóng xuống: ứng viên cỡ k có đúng k tập con cỡ $k-1$, mọi tập con này phải nằm trong L_{k-1} .

The join step takes two sets of L_{k-1} that share $k-2$ items and unites them into a set of size k . Implementations usually order the items and join only two sets that agree on their first $k-2$ items; this generates each candidate once and, after pruning, gives the same C_k as joining every pair. The prune step is where downward closure is used: a candidate of size k has exactly k subsets of size $k-1$, and all of them must be in L_{k-1} .

```
L1 = các mục có số đếm ≥ minsup
với k = 2, 3, ... trong khi L(k-1) khác rỗng:
    Ck = ghép(L(k-1), L(k-1))
    xóa khỏi Ck mọi tập có tập con cỡ k-1 không thuộc L(k-1)
    quét CSDL, đếm từng tập trong Ck
    Lk = các tập trong Ck có số đếm ≥ minsup
kết quả = hợp của L1, L2, ...
```

```
L1 = items whose count ≥ minsup
for k = 2, 3, ... while L(k-1) is not empty:
    Ck = join(L(k-1), L(k-1))
    delete from Ck every set with a (k-1)-subset not in L(k-1)
    scan the database and count each set in Ck
    Lk = sets in Ck whose count ≥ minsup
result = union of L1, L2, ...
```

Mã sau minh họa ghép và tĩa ở mức $k = 3$ của dữ liệu Phú Sa, bắt đầu từ bốn cặp phổ biến L_2 (kết quả mục 4.4).

The code that follows illustrates join and prune at level $k = 3$ of the Phú Sa data, starting from the four frequent pairs of L_2 (results in section 4.4).

```

from itertools import combinations
ORD = "TD VM DM XM BH".split()
L2 = [("TD", "VM"), ("TD", "BH"), ("VM", "BH"), ("XM", "BH")]
ghep = []
for a, b in combinations(L2, 2):
    u = tuple(sorted(set(a) | set(b), key=ORD.index))
    if len(u) == 3 and u not in ghep:
        ghep.append(u)
print("Ghép từ L2:", len(ghep), "ứng viên")
for u in ghep:
    thieu = [c for c in combinations(u, 2) if c not in L2]
    if thieu:
        print(" ".join(u), "-> loại, thiếu", thieu)
    else:
        print(" ".join(u), "-> giữ")

```

Kết quả chạy / Output

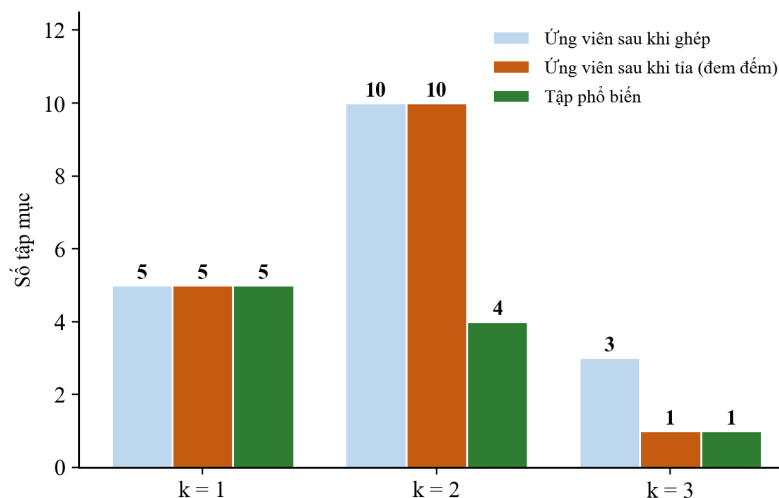
```

Ghép từ L2: 3 ứng viên
TD VM BH -> giữ
TD XM BH -> loại, thiếu [('TD', 'XM')]
VM XM BH -> loại, thiếu [('VM', 'XM')]

```

Ghép các cặp của L2 cho ba ứng viên. Hai ứng viên $\{TD, XM, BH\}$ và $\{VM, XM, BH\}$ bị tĩa vì chứa cặp $\{TD, XM\}$ và $\{VM, XM\}$ không phổ biến, nên không cần đếm. Chỉ $\{TD, VM, BH\}$ đi tiếp. Hình 4.2 tóm tắt số ứng viên ở từng mức.

Joining the pairs of L2 gives three candidates. The candidates $\{TD, XM, BH\}$ and $\{VM, XM, BH\}$ are pruned because they contain the infrequent pairs $\{TD, XM\}$ and $\{VM, XM\}$, so they are never counted. Only $\{TD, VM, BH\}$ goes on. Figure 4.2 summarises the number of candidates at each level.



Hình 4.2. Số ứng viên và số tập phổ biến theo mức k: tĩa làm số tập cần đếm ở k = 3 giảm từ 3 xuống 1

Figure 4.2. Number of candidates and frequent itemsets by level k: pruning cuts the sets to count at k = 3 from 3 to 1

PHẢI THUỘC / MEMORISE — Mọi tập con của tập phổ biến đều phổ biến. Ứng viên cỡ k chỉ được đếm khi cả k tập con cỡ k-1 của nó nằm trong L_{k-1} . Apriori quét cơ sở dữ liệu một lần cho mỗi mức k.

Every subset of a frequent itemset is frequent. A candidate of size k is counted only when all k of its (k-1)-subsets are in L_{k-1} . Apriori scans the database once per level k.

LỖI HAY GẶP / COMMON MISTAKE — Tỉa thiếu: chỉ kiểm hai tập đã dùng để ghép mà bỏ qua các tập con còn lại. Ứng viên cỡ k có k tập con cỡ $k-1$; phải kiểm đủ k tập. Ngược lại, tỉa nhầm theo chiều xuôi (loại tập con vì tập lớn không phổ biến) là sai, vì tập con của tập không phổ biến vẫn có thể phổ biến.

Incomplete pruning: checking only the two sets used for the join and skipping the other subsets. A candidate of size k has k subsets of size $k-1$, and all k must be checked. Pruning in the wrong direction (dropping a subset because the larger set is infrequent) is also wrong, because a subset of an infrequent set can still be frequent.

4.4. Bài mẫu 1: chạy tay Apriori trên giao dịch sản phẩm của Phú Sa — 4.4.

Worked example 1: Apriori by hand on the Phú Sa product transactions

Đề bài: cho 10 giao dịch GD1–GD10 như mục 4.1 và minsup = 30%. Tìm mọi tập phổ biến bằng Apriori. Vì $n = 10$ nên minsup tương ứng số đếm tối thiểu 3; tập có số đếm 3 vẫn phổ biến vì ngưỡng là "không nhỏ hơn".

Problem: take the 10 transactions GD1–GD10 of section 4.1 and minsup = 30%. Find all frequent itemsets with Apriori. Since $n = 10$, minsup corresponds to a minimum count of 3; a set with count 3 is still frequent because the threshold is "at least".

Bước 1. C1 và L1 — Step 1. C1 and L1

C1 là 5 mục đơn. Quét một lần để đếm: không mục nào bị loại, nên $L1 = C1$ (hai sản phẩm trả góp có đếm đúng bằng 3).

C1 is the 5 single items. One scan counts them: no item is dropped, so $L1 = C1$ (the two instalment products have a count of exactly 3).

Mục Item	Đếm Count	Support	Kết luận Verdict
TD	6	60%	Giữ (đếm ≥ 3) Keep (count ≥ 3)
VM	6	60%	Giữ (đếm ≥ 3) Keep (count ≥ 3)
DM	3	30%	Giữ (đếm ≥ 3) Keep (count ≥ 3)
XM	3	30%	Giữ (đếm ≥ 3) Keep (count ≥ 3)
BH	8	80%	Giữ (đếm ≥ 3) Keep (count ≥ 3)

Bước 2. C2 và L2 — Step 2. C2 and L2

Ghép mọi cặp của L1 cho C2 gồm $C(5, 2) = 10$ cặp. Các mục đều phổ biến nên chưa có cặp nào bị tỉa. Quét lần hai để đếm. Chỉ bốn cặp đạt ngưỡng: {TD, VM}, {TD, BH}, {VM, BH}, {XM, BH}.

Joining every pair of $L1$ gives $C2$ with $C(5, 2) = 10$ pairs. All items are frequent, so no pair is pruned yet. A second scan counts them. Only four pairs reach the threshold: $\{TD, VM\}$, $\{TD, BH\}$, $\{VM, BH\}$, $\{XM, BH\}$.

Cặp <i>Pair</i>	Đếm <i>Count</i>	Support	Kết luận <i>Verdict</i>
TD + VM	4	40%	Giữ vào L2 <i>Keep in L2</i>
TD + DM	1	10%	Loại (đếm < 3) <i>Drop (count < 3)</i>
TD + XM	0	0%	Loại (đếm < 3) <i>Drop (count < 3)</i>
TD + BH	4	40%	Giữ vào L2 <i>Keep in L2</i>
VM + DM	1	10%	Loại (đếm < 3) <i>Drop (count < 3)</i>
VM + XM	1	10%	Loại (đếm < 3) <i>Drop (count < 3)</i>
VM + BH	5	50%	Giữ vào L2 <i>Keep in L2</i>
DM + XM	1	10%	Loại (đếm < 3) <i>Drop (count < 3)</i>
DM + BH	2	20%	Loại (đếm < 3) <i>Drop (count < 3)</i>
XM + BH	3	30%	Giữ vào L2 <i>Keep in L2</i>

Bước 3. C3 và L3 — Step 3. C3 and L3

Ghép các cặp của $L2$ cho ba ứng viên cỡ 3. Kiểm ba cặp con của mỗi ứng viên: hai ứng viên chứa cặp không thuộc $L2$ nên bị tĩa. Ứng viên còn lại $\{TD, VM, BH\}$ có đủ ba cặp con trong $L2$; quét lần ba cho số đếm 3 (GD1, GD4, GD10), đạt ngưỡng.

Joining the pairs of $L2$ gives three candidates of size 3. Checking the three pair-subsets of each candidate, two candidates contain a pair outside $L2$ and are pruned. The remaining candidate $\{TD, VM, BH\}$ has all three pair-subsets in $L2$; the third scan gives a count of 3 (GD1, GD4, GD10), which meets the threshold.

Ghép từ <i>Joined from</i>	Ứng viên <i>Candidate</i>	Cặp con thiếu trong L2 <i>Subsets missing from L2</i>	Kết quả <i>Result</i>
TD + VM ; TD + BH	TD + VM + BH	không có <i>none</i>	Giữ, đem đếm <i>Keep and count</i>
TD + BH ; XM + BH	TD + XM + BH	{TD XM}	Tỉa <i>Pruned</i>
VM + BH ; XM + BH	VM + XM + BH	{VM XM}	Tỉa <i>Pruned</i>

Bước 4. Dừng — *Step 4. Stop*

L3 chỉ có một tập nên không ghép được cặp nào để sinh C4; thuật toán dừng. Tổng cộng có $5 + 4 + 1 = 10$ tập phổ biến. Số tập phải đếm là $5 + 10 + 1 = 16$, so với 31 tập mục có thể có. Mã sau chạy lại toàn bộ quá trình.

L3 has a single set, so no pair can be joined to generate C4; the algorithm stops. There are $5 + 4 + 1 = 10$ frequent itemsets in all. The number of sets to count is $5 + 10 + 1 = 16$, against 31 possible itemsets. The code that follows reruns the whole process.

```
import textwrap
from itertools import combinations
GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,TD VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
ORD = "TD VM DM XM BH".split()
MIN = 3 # 30% của 10 giao dịch

def dem(c):
    return sum(1 for t in GD if set(c) <= t)

def in_ra(nhan, ds):
    print(textwrap.fill(nhan + " " + ", ".join(ds), 70,
                        subsequent_indent="    "))

C, k = [(i,) for i in ORD], 1
while C:
    cnt = {c: dem(c) for c in C}
    L = [c for c in C if cnt[c] >= MIN]
    in_ra(f"C{k}:", [f"{'+'.join(c)}={cnt[c]}" for c in C])
    in_ra(f"L{k}:", ["+".join(c) for c in L])
    C = []
    for a, b in combinations(L, 2):
        u = tuple(sorted(set(a) | set(b), key=ORD.index))
        if (len(u) == k + 1 and u not in C
            and all(s in L for s in combinations(u, k))):
            C.append(u)
    k += 1
```

Kết quả chạy / *Output*

```
C1: TD=6, VM=6, DM=3, XM=3, BH=8
L1: TD, VM, DM, XM, BH
C2: TD+VM=4, TD+DM=1, TD+XM=0, TD+BH=4, VM+DM=1, VM+XM=1, VM+BH=5,
    DM+XM=1, DM+BH=2, XM+BH=3
L2: TD+VM, TD+BH, VM+BH, XM+BH
C3: TD+VM+BH=3
L3: TD+VM+BH
```

Kết quả máy khớp bảng tay. Số đếm của các tập phổ biến: {TD} 6, {VM} 6, {DM} 3, {XM} 3, {BH} 8; {TD, VM} 4, {TD, BH} 4, {VM, BH} 5, {XM, BH} 3; {TD, VM, BH} 3. Chia cho 10 ta có độ hỗ trợ 0,6; 0,6; 0,3; 0,3; 0,8; 0,4; 0,4; 0,5; 0,3; 0,3. Thư viện mlxtend của Python có hàm apriori và fpgrowth làm việc này trên bảng pandas, và Weka có lớp Apriori; cần chọn ngưỡng và đọc kết quả theo cùng quy tắc như trên.

The machine result matches the hand tables. Counts of the frequent sets: {TD} 6, {VM} 6, {DM} 3, {XM} 3, {BH} 8; {TD, VM} 4, {TD, BH} 4, {VM, BH} 5, {XM, BH} 3; {TD, VM, BH} 3. Dividing by 10 gives supports 0.6, 0.6, 0.3, 0.3, 0.8, 0.4, 0.4, 0.5, 0.3, 0.3. The Python library mlxtend has functions apriori and fpgrowth that do this on a pandas table, and Weka has an Apriori class; the thresholds must be chosen and the results read by the same rules as above.

LỖI HAY GẶP / COMMON MISTAKE — Loại tập có số đếm đúng bằng ngưỡng. Với minsup = 30% và n = 10, {DM}, {XM}, {XM, BH} và {TD, VM, BH} đều có số đếm 3 và vẫn phổ biến; dùng dấu lớn hơn thay vì lớn hơn hoặc bằng làm mất cả bốn tập.

Dropping a set whose count equals the threshold. With minsup = 30% and n = 10, {DM}, {XM}, {XM, BH} and {TD, VM, BH} all have a count of 3 and remain frequent; using a strict greater-than instead of greater-or-equal loses all four.

LỖI HAY GẶP / COMMON MISTAKE — Không tỉa mà đếm thẳng mọi ứng viên cỡ 3. Ở bài này không tỉa vẫn ra cùng L3, nhưng phải đếm thêm hai tập vô ích; với dữ liệu thật hàng triệu giao dịch, mỗi ứng viên thừa là một lần so khớp thừa trên toàn bộ cơ sở dữ liệu.

Counting every size-3 candidate without pruning. Here the result is still the same L3, but two useless sets are counted; with real data of millions of transactions, each surplus candidate is a surplus match against the whole database.

4.5. Sinh luật từ tập phổ biến và lọc theo độ tin cậy — 4.5. Generating rules from frequent itemsets and filtering by confidence

Khai phá luật kết hợp gồm hai giai đoạn: (1) tìm mọi tập phổ biến, việc tốn kém đã làm ở mục 4.4; (2) từ mỗi tập phổ biến F có từ hai mục trở lên, sinh các luật $A \Rightarrow F - A$ với A là tập con khác rỗng và khác F, rồi giữ luật đạt minconf. Luật sinh từ F tự đạt minsup vì độ hỗ trợ của luật bằng độ hỗ trợ của F. Số đếm của F và của A đã có sẵn từ giai đoạn một nhờ tính chất đóng xuống (mọi tập con của F đều phổ biến), nên giai đoạn hai không cần quét lại cơ sở dữ liệu.

Association rule mining has two stages: (1) find all frequent itemsets, the expensive work done in section 4.4; (2) from each frequent set F with two or more items, generate rules $A \Rightarrow F - A$ where A is a non-empty proper subset of F, and keep those that meet minconf. A rule generated from F meets minsup automatically because its support equals the support of F. The counts of F and of A are already available from stage one thanks to downward closure (every subset of F is frequent), so stage two needs no further scan.

$$\text{Số luật sinh từ một tập phổ biến } k \text{ mục} = 2^k - 2 \quad (4.9)$$

Tập hai mục cho 2 luật, tập ba mục cho 6 luật. Độ tin cậy của luật $A \Rightarrow F - A$ bằng số đếm của F chia số đếm của A. Có một cách tỉa luật: nếu $A \Rightarrow F - A$ không đạt minconf thì mọi luật $A' \Rightarrow F - A'$ với A' là tập con của A cũng không đạt, vì số đếm của A' không

nhỏ hơn số đếm của A nên mẫu số chỉ lớn lên. Chuyển mục từ về trái sang về phải không bao giờ làm độ tin cậy tăng.

A set of two items gives 2 rules and a set of three items gives 6. The confidence of $A \Rightarrow F - A$ is the count of F divided by the count of A. Rules can be pruned: if $A \Rightarrow F - A$ fails minconf, then every rule $A' \Rightarrow F - A'$ with A' a subset of A fails too, because the count of A' is not smaller than the count of A, so the denominator only grows. Moving an item from the left side to the right side never raises confidence.

Bài mẫu 2. Từ các tập phổ biến ở mục 4.4 và minconf = 70%, sinh mọi luật kết hợp mạnh. Có 4 tập hai mục và 1 tập ba mục, nên có $4 \times 2 + 6 = 14$ luật ứng viên. Bảng sau liệt kê từng luật với số đếm của F, số đếm của vế trái và độ tin cậy.

Worked example 2. From the frequent sets of section 4.4 and minconf = 70%, generate all strong association rules. There are 4 two-item sets and 1 three-item set, hence $4 \times 2 + 6 = 14$ candidate rules. The next table lists each rule with the count of F, the count of the left side and the confidence.

Luật Rule	Đếm F / đếm vế trái Count F / left side	Confidence	Kết luận Verdict
TD $\Rightarrow$ VM	4 / 6	0,6667	Loại Reject
VM $\Rightarrow$ TD	4 / 6	0,6667	Loại Reject
TD $\Rightarrow$ BH	4 / 6	0,6667	Loại Reject
BH $\Rightarrow$ TD	4 / 8	0,5000	Loại Reject
VM $\Rightarrow$ BH	5 / 6	0,8333	Đạt ($\geq 0,7$) Pass (≥ 0.7)
BH $\Rightarrow$ VM	5 / 8	0,6250	Loại Reject
XM $\Rightarrow$ BH	3 / 3	1,0000	Đạt ($\geq 0,7$) Pass (≥ 0.7)
BH $\Rightarrow$ XM	3 / 8	0,3750	Loại Reject
TD $\Rightarrow$ VM, BH	3 / 6	0,5000	Loại Reject
VM $\Rightarrow$ TD, BH	3 / 6	0,5000	Loại Reject
BH $\Rightarrow$ TD, VM	3 / 8	0,3750	Loại Reject

Luật <i>Rule</i>	Đếm F / đếm vế trái <i>Count F / left side</i>	Confidence	Kết luận <i>Verdict</i>
TD, VM $\Rightarrow$ BH	3 / 4	0,7500	Đạt ($\geq 0,7$) <i>Pass (≥ 0.7)</i>
TD, BH $\Rightarrow$ VM	3 / 4	0,7500	Đạt ($\geq 0,7$) <i>Pass (≥ 0.7)</i>
VM, BH $\Rightarrow$ TD	3 / 5	0,6000	Loại <i>Reject</i>

```

from itertools import combinations
GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
MIN_CONF = 0.7

def dem(c):
    return sum(1 for t in GD if set(c) <= t)

L = [("TD", "VM"), ("TD", "BH"), ("VM", "BH"), ("XM", "BH"),
      ("TD", "VM", "BH")]
for tap in L:
    for r in range(1, len(tap)):
        for a in combinations(tap, r):
            b = tuple(x for x in tap if x not in a)
            conf = dem(tap) / dem(a)
            kq = "đạt" if conf >= MIN_CONF else "loại"
            print(f'{', '.join(a):<6} => {', '.join(b):<6}'
                  f' {dem(tap)}/{dem(a)} = {conf:.4f} {kq}')
```

Kết quả chạy / Output

```

TD      => VM      4/6 = 0.6667 loại
VM      => TD      4/6 = 0.6667 loại
TD      => BH      4/6 = 0.6667 loại
BH      => TD      4/8 = 0.5000 loại
VM      => BH      5/6 = 0.8333 đạt
BH      => VM      5/8 = 0.6250 loại
XM      => BH      3/3 = 1.0000 đạt
BH      => XM      3/8 = 0.3750 loại
TD      => VM,BH   3/6 = 0.5000 loại
VM      => TD,BH   3/6 = 0.5000 loại
BH      => TD,VM   3/8 = 0.3750 loại
TD,VM   => BH      3/4 = 0.7500 đạt
TD,BH   => VM      3/4 = 0.7500 đạt
VM,BH   => TD      3/5 = 0.6000 loại
```

Bốn luật mạnh: VM $\Rightarrow$ BH (độ tin cậy 0,8333), XM $\Rightarrow$ BH (1,0), TD, VM $\Rightarrow$ BH (0,75) và TD, BH $\Rightarrow$ VM (0,75). Hai luật cuối sinh từ {TD, VM, BH}; bốn luật còn lại của tập này bị loại, trong đó BH $\Rightarrow$ TD, VM có 0,375 và VM, BH $\Rightarrow$ TD có 0,6. Kết quả khớp với danh sách luật mạnh chuẩn của bộ dữ liệu, với độ hỗ trợ lần lượt 0,5; 0,3; 0,3; 0,3.

Four strong rules: VM $\Rightarrow$ BH (confidence 0.8333), XM $\Rightarrow$ BH (1.0), TD, VM $\Rightarrow$ BH (0.75) and TD, BH $\Rightarrow$ VM (0.75). The last two come from {TD, VM, BH}; the other four rules of this set are rejected, among them BH $\Rightarrow$ TD, VM with 0.375 and VM, BH $\Rightarrow$ TD with 0.6. The result matches the standard list of strong rules of the data set, with supports of 0.5, 0.3, 0.3 and 0.3.

PHẢI THUỘC / MEMORISE — Luật sinh từ tập phổ biến $F: A \Rightarrow F - A$ với mọi tập con A khác rỗng, khác F . Độ tin cậy = $\text{đếm}(F) / \text{đếm}(A)$. Giữ luật có độ tin cậy không nhỏ hơn minconf ; không cần quét lại dữ liệu.

Rules come from a frequent set $F: A \Rightarrow F - A$ for every non-empty proper subset A . Confidence = $\text{count}(F) / \text{count}(A)$. Keep rules whose confidence is at least minconf ; no rescan of the data is needed.

LỖI HAY GẶP / COMMON MISTAKE — Lấy số đếm của F chia cho tổng số giao dịch rồi gọi đó là độ tin cậy. Đó là độ hỗ trợ: $TD, VM \Rightarrow BH$ có $3 / 10 = 0,3$ là support, còn confidence là $3 / 4 = 0,75$ vì mẫu số là số giao dịch có TD và VM .

Dividing the count of F by the total number of transactions and calling it confidence. That is support: for $TD, VM \Rightarrow BH$, $3 / 10 = 0.3$ is the support, while the confidence is $3 / 4 = 0.75$ because the denominator is the number of transactions holding TD and VM .

4.6. Luật mạnh nhưng không thú vị: đánh giá bằng lift, leverage và conviction

— 4.6. Strong but uninteresting rules: evaluation with lift, leverage and conviction

Bài mẫu 3. Bốn luật mạnh ở mục 4.5 đều qua ngưỡng support và confidence, nhưng chưa chắc đáng đưa vào chiến dịch bán chéo. Bảng sau thêm lift, leverage và conviction cho từng luật; mức nền của vế phải là 0,8 với BH và 0,6 với VM .

Worked example 3. The four strong rules of section 4.5 all pass the support and confidence thresholds, but they may not deserve a place in a cross-selling campaign. The next table adds lift, leverage and conviction for each rule; the base rate of the right side is 0.8 for BH and 0.6 for VM .

Luật Rule	Support	Confidence	Lift	Leverage	Conviction
$VM \Rightarrow BH$	0,5	0,8333	1,0417	0,02	1,2000
$XM \Rightarrow BH$	0,3	1,0000	1,2500	0,06	∞
$TD, BH \Rightarrow VM$	0,3	0,7500	1,2500	0,06	1,6000
$TD, VM \Rightarrow BH$	0,3	0,7500	0,9375	-0,02	0,8000

```

GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
n = len(GD)

def dem(c):
    return sum(1 for t in GD if set(c) <= t)

LUAT = [("VM", "BH"), ("XM", "BH"), ("TD BH", "VM"), ("TD VM", "BH")]
for vt, vp in LUAT:
    a, b = set(vt.split()), set(vp.split())
    conf = dem(a | b) / dem(a)
    nen = dem(b) / n
    lift = conf / nen
    loai = "duong" if lift > 1 else "âm" if lift < 1 else "độc lập"
    print(f"{vt} => {vp}: conf {conf:.4f}, nền {nen:.1f},"
          f" lift {lift:.4f} ({loai})")

```

Kết quả chạy / Output

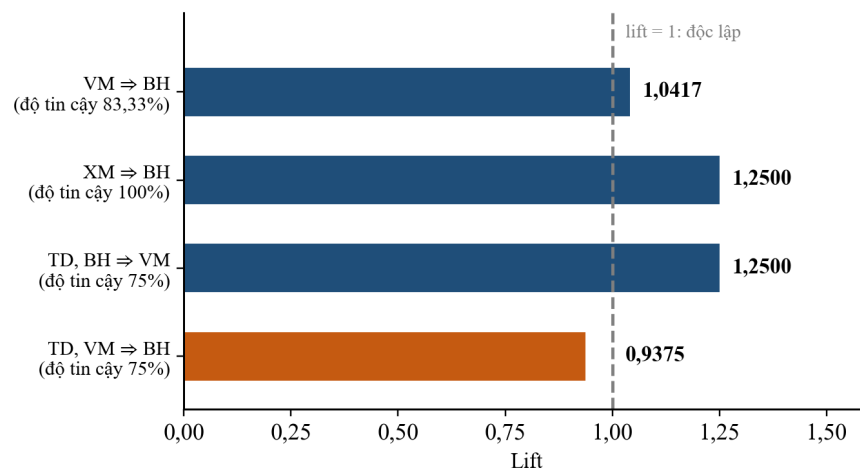
```

VM => BH: conf 0.8333, nền 0.8, lift 1.0417 (dương)
XM => BH: conf 1.0000, nền 0.8, lift 1.2500 (dương)
TD BH => VM: conf 0.7500, nền 0.6, lift 1.2500 (dương)
TD VM => BH: conf 0.7500, nền 0.8, lift 0.9375 (âm)

```

Đọc kết quả theo từng luật. Luật TD, VM $\Rightarrow$ BH có độ tin cậy 0,75 nhưng thấp hơn mức nền 0,8: lift 0,9375, leverage -0,02, conviction 0,8; đây là luật mạnh mà tương quan âm, không nên dùng để mời mua bảo hiểm. Luật VM $\Rightarrow$ BH có lift 1,0417, gần với độc lập; độ tin cậy 83,33% phần lớn đến từ việc 80% giao dịch vốn có bảo hiểm. Luật TD, BH $\Rightarrow$ VM và XM $\Rightarrow$ BH có lift 1,25 và leverage 0,06, là hai luật thú vị nhất. Tuy vậy XM $\Rightarrow$ BH dựa trên 3 giao dịch, nên độ tin cậy 100% có thể chỉ là ngẫu nhiên của mẫu nhỏ.

Reading the result rule by rule. Rule TD, VM $\Rightarrow$ BH has confidence 0.75 but under the base rate of 0.8: lift 0.9375, leverage -0.02, conviction 0.8; it is a strong rule with negative correlation and should not be used to invite insurance purchases. Rule VM $\Rightarrow$ BH has lift 1.0417, close to independence; its confidence of 83.33% comes largely from the fact that 80% of transactions already carry insurance. Rules TD, BH $\Rightarrow$ VM and XM $\Rightarrow$ BH have lift 1.25 and leverage 0.06 and are the most interesting. Yet XM $\Rightarrow$ BH rests on 3 transactions, so its 100% confidence may be chance in a small sample.



Hình 4.3. Lift của bốn luật mạnh: cột xanh là tương quan dương, cột cam là tương quan âm, đường nét đứt là lift bằng 1

Figure 4.3. Lift of the four strong rules: blue bars show positive correlation, orange bars negative correlation, the dashed line is lift = 1

Lift, leverage và conviction đều lấy tính độc lập làm điểm gốc, nhưng khác cách đo. Lift là tỷ số nên nhạy với tập hiếm; leverage là hiệu nên cho biết quy mô tuyệt đối (bao nhiêu phần giao dịch lệch khỏi kỳ vọng); conviction có hướng, khác lift ở chỗ không đối xứng. Không có ngưỡng lift phổ quát: ngưỡng phụ thuộc chi phí của một lời mời sai và lợi nhuận của một lời mời đúng. Ngoài ba độ đo này còn có kiểm định χ^2 cho bảng 2×2 của hai tập mục, dùng khi cần kết luận thống kê về sự phụ thuộc.

Lift, leverage and conviction all take independence as the reference point but measure differently. Lift is a ratio and hence sensitive to rare sets; leverage is a difference and shows the absolute scale (what share of transactions departs from expectation); conviction is directional,

unlike lift, which is symmetric. There is no universal lift threshold: it depends on the cost of a wrong offer and the profit of a right one. Besides these three there is the χ^2 test on the 2×2 table of two itemsets, used when a statistical conclusion about dependence is needed.

LỖI HAY GẶP / COMMON MISTAKE — Chọn luật chỉ theo độ tin cậy cao nhất. Với dữ liệu Phú Sa, luật $XM \Rightarrow BH$ đứng đầu về confidence (100%) nhưng chỉ có 3 giao dịch, còn $TD, VM \Rightarrow BH$ vượt ngưỡng 70% mà lift nhỏ hơn 1. Luôn đọc confidence cùng lift và support.

Choosing rules by highest confidence alone. In the Phú Sa data $XM \Rightarrow BH$ leads on confidence (100%) but has only 3 transactions, and $TD, VM \Rightarrow BH$ passes the 70% threshold with a lift under 1. Always read confidence together with lift and support.

LỖI HAY GẶP / COMMON MISTAKE — Kết luận lift lớn hơn 1 nghĩa là X gây ra Y. Lift chỉ nói hai tập hay đi cùng nhau hơn kỳ vọng, không nói thứ tự hay nguyên nhân; thẻ tín dụng không "làm" khách mua thêm vay tiền mặt.

Concluding that lift above 1 means X causes Y. Lift only says that two sets go together more than expected; it says nothing about order or cause, and the credit card does not "make" customers take a cash loan.

4.7. Tập phổ biến đóng và tập phổ biến cực đại — 4.7. Closed and maximal frequent itemsets

Khi minsup thấp, số tập phổ biến bùng nổ vì mọi tập con của một tập phổ biến lớn đều phổ biến: một tập 20 mục phổ biến kéo theo hơn một triệu tập con. Hai cách biểu diễn gọn hơn là tập đóng và tập cực đại.

When minsup is low the number of frequent itemsets explodes, because every subset of a large frequent set is frequent: one frequent set of 20 items brings more than a million subsets. Two more compact representations are the closed and the maximal itemsets.

Khái niệm Concept	Định nghĩa Definition	Thông tin giữ lại Information kept
Tập phổ biến đóng <i>Closed frequent itemset</i>	Tập phổ biến X mà không có tập chứa X nào khác X (tập cha) có cùng số đếm <i>A frequent set X with no proper superset of the same count</i>	Đủ để suy ra support của mọi tập phổ biến <i>Enough to recover the support of every frequent set</i>
Tập phổ biến cực đại <i>Maximal frequent itemset</i>	Tập phổ biến X mà không có tập chứa X nào khác X (tập cha) mà phổ biến <i>A frequent set X with no frequent proper superset</i>	Chỉ biết ranh giới của vùng phổ biến, mất support chính xác <i>Only the boundary of the frequent region; exact supports are lost</i>

$$\text{support}(X) = \max \{ \text{support}(C) : C \text{ là tập đóng, } X \subseteq C \} \quad (4.10)$$

Quan hệ bao hàm: tập cực đại $\subseteq$ tập đóng $\subseteq$ tập phổ biến. Bài mẫu 4: phân loại 10 tập phổ biến của mục 4.4. Với mỗi tập, tìm tập cha có cùng số đếm (để kết luận có đóng không) và tập cha phổ biến (để kết luận có cực đại không).

Inclusion: maximal sets $\subseteq$ closed sets $\subseteq$ frequent sets. Worked example 4: classify the 10 frequent sets of section 4.4. For each set, look for a superset of the same count (to decide whether it is closed) and for a frequent superset (to decide whether it is maximal).

Tập Set	Đếm Count	Đóng? Closed?	Cực đại? Maximal?
{TD}	6	Có <i>Yes</i>	Không: có {TD, BH} <i>No: {TD, BH} is frequent</i>
{VM}	6	Có <i>Yes</i>	Không: có {TD, VM} <i>No: {TD, VM} is frequent</i>
{DM}	3	Có <i>Yes</i>	Có <i>Yes</i>
{XM}	3	Không: {XM, BH} cùng đếm 3 <i>No: {XM, BH} has the same count 3</i>	Không: có {XM, BH} <i>No: {XM, BH} is frequent</i>
{BH}	8	Có <i>Yes</i>	Không: có {TD, BH} <i>No: {TD, BH} is frequent</i>
{TD, VM}	4	Có <i>Yes</i>	Không: có {TD, VM, BH} <i>No: {TD, VM, BH} is frequent</i>
{TD, BH}	4	Có <i>Yes</i>	Không: có {TD, VM, BH} <i>No: {TD, VM, BH} is frequent</i>
{VM, BH}	5	Có <i>Yes</i>	Không: có {TD, VM, BH} <i>No: {TD, VM, BH} is frequent</i>
{XM, BH}	3	Có <i>Yes</i>	Có <i>Yes</i>
{TD, VM, BH}	3	Có <i>Yes</i>	Có <i>Yes</i>

```

from itertools import combinations
GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH, "
       "TD VM,DM XM BH,TD VM BH").split(",")]
ORD = "TD VM DM XM BH".split()
MIN = 3
tat = {}
for r in range(1, 6):
    for c in combinations(ORD, r):
        d = sum(1 for t in GD if set(c) <= t)
        if d >= MIN:
            tat[c] = d
dong = [c for c in tat
        if not any(set(c) < set(x) and tat[x] == tat[c] for x in tat)]
cucdai = [c for c in tat if not any(set(c) < set(x) for x in tat)]
print("Phổ biến:", len(tat), "| đóng:", len(dong),
      "| cực đại:", len(cucdai))
print("Không đóng:", [c for c in tat if c not in dong])
for c in cucdai:
    print("Cực đại:", " ".join(c), tat[c])

```

Kết quả chạy / Output

```

Phổ biến: 10 | đóng: 9 | cực đại: 3
Không đóng: [('XM',)]
Cực đại: DM 3
Cực đại: XM BH 3
Cực đại: TD VM BH 3

```

Trong 10 tập phổ biến có 9 tập đóng; chỉ {XM} không đóng vì {XM, BH} có cùng số đếm 3, nghĩa là mọi giao dịch có XM đều có BH. Có 3 tập cực đại: {DM}, {XM, BH} và {TD, VM, BH}. Từ tập đóng suy ra support của {XM}: tập đóng nhỏ nhất chứa {XM} là {XM, BH} với số đếm 3. Từ ba tập cực đại thì chỉ biết {TD, VM} phổ biến, không biết số đếm của nó là 4.

Of the 10 frequent sets, 9 are closed; only {XM} is not, because {XM, BH} has the same count of 3, meaning every transaction with XM also has BH. There are 3 maximal sets: {DM}, {XM, BH} and {TD, VM, BH}. From the closed sets the support of {XM} is recovered: the smallest closed set containing {XM} is {XM, BH} with count 3. From the three maximal sets one only knows that {TD, VM} is frequent, not that its count is 4.

LỖI HAY GẶP / COMMON MISTAKE — Nhầm tập đóng với tập cực đại. Tập cực đại luôn đóng, nhưng tập đóng chưa chắc cực đại: {TD, VM} là tập đóng (số đếm 4, cha {TD, VM, BH} chỉ có 3) mà không cực đại vì cha của nó phổ biến.

Confusing closed with maximal. A maximal set is always closed, but a closed set need not be maximal: {TD, VM} is closed (count 4, its superset {TD, VM, BH} has only 3) but not maximal because that superset is frequent.

4.8. Thuật toán FP-growth và cây FP-tree — 4.8. The FP-growth algorithm and the FP-tree

Apriori phải sinh và đếm nhiều ứng viên, và quét cơ sở dữ liệu một lần mỗi mức. FP-growth (frequent pattern growth) tránh cả hai bằng cách nén cơ sở dữ liệu vào một cây tiền tố gọi là FP-tree, rồi khai phá trực tiếp trên cây. Thuật toán chỉ quét cơ sở dữ liệu hai lần.

Apriori has to generate and count many candidates and scans the database once per level. FP-growth (frequent pattern growth) avoids both by compressing the database into a prefix tree called the FP-tree and mining the tree directly. The algorithm scans the database only twice.

Giai đoạn 1: dựng cây. Quét lần một đếm tần số từng mục và bỏ mục không phổ biến (ở đây cả 5 mục đều phổ biến). Sắp các mục theo tần số giảm dần; khi hai mục bằng tần số thì theo thứ tự TD, VM, DM, XM, BH đã quy ước. Kết quả: BH (8), TD (6), VM (6), DM (3), XM (3). Quét lần hai sắp từng giao dịch theo thứ tự này (bảng sau) rồi chèn vào cây từ gốc: chung tiền tố thì tăng số đếm của nút, khác thì mở nhánh mới. Bảng tiêu đề (header table) giữ danh sách liên kết các nút cùng mục, dùng ở giai đoạn khai phá.

Stage 1: build the tree. The first scan counts the frequency of each item and drops infrequent items (here all 5 are frequent). Items are sorted by descending frequency; when two items tie, the agreed order TD, VM, DM, XM, BH decides. The result is BH (8), TD (6), VM (6), DM (3), XM (3). The second scan sorts each transaction in this order (next table) and inserts it into the tree

from the root: a shared prefix raises the count of the node, a different item opens a new branch.
The header table keeps a linked list of the nodes of each item for use in the mining stage.

Giao dịch <i>Transaction</i>	Tập mục <i>Itemset</i>	Đã sắp theo tần số giảm <i>Sorted by descending frequency</i>
GD1	{TD, VM, BH}	BH, TD, VM
GD2	{TD, BH}	BH, TD
GD3	{VM, DM, BH}	BH, VM, DM
GD4	{TD, VM, BH}	BH, TD, VM
GD5	{XM, BH}	BH, XM
GD6	{TD, DM}	TD, DM
GD7	{VM, XM, BH}	BH, VM, XM
GD8	{TD, VM}	TD, VM
GD9	{DM, XM, BH}	BH, DM, XM
GD10	{TD, VM, BH}	BH, TD, VM

```

GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH, "
       "TD VM,DM XM BH,TD VM BH").split(",")]
ORD = "TD VM DM XM BH".split()
tan = {i: sum(1 for t in GD if i in t) for i in ORD}
thu_tu = sorted(tan, key=lambda i: (-tan[i], ORD.index(i)))
print("Thứ tự mục:", thu_tu)
cay = {}
for t in GD:
    nut = cay
    for i in [x for x in thu_tu if x in t]:
        nut.setdefault(i, [0, {}])
        nut[i][0] += 1
        nut = nut[i][1]

def in_cay(nut, cap=0):
    for i, (n, con) in nut.items():
        print(" " * cap + f"{i}:{n}")
        in_cay(con, cap + 1)

in_cay(cay)

```

Kết quả chạy / Output

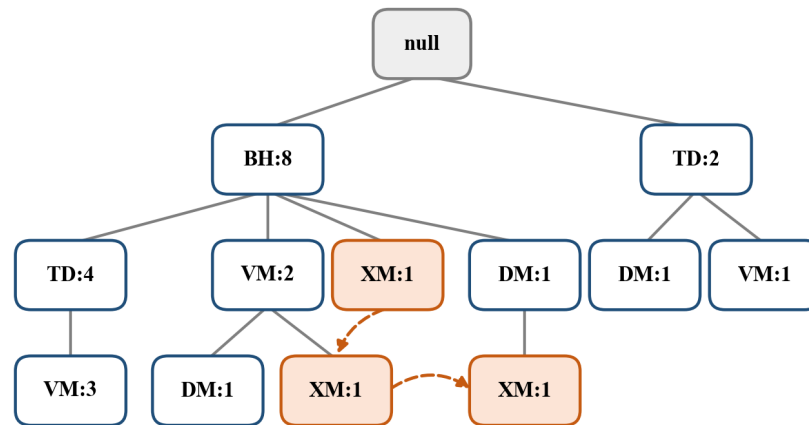
```

Thứ tự mục: ['BH', 'TD', 'VM', 'DM', 'XM']
BH:8
  TD:4
    VM:3
      VM:2
        DM:1
          XM:1
            XM:1
              DM:1
                XM:1
      TD:2
        DM:1
        VM:1

```

Cây có 12 nút (không tính gốc) thay cho 26 lần xuất hiện của mục trong 10 giao dịch. Nút BH:8 là tiền tố chung của 8 giao dịch; nhánh TD:2 bên phải chứa hai giao dịch không có BH (GD6 và GD8). Hình 4.4 vẽ cây này kèm các liên kết của mục XM.

The tree has 12 nodes (root excluded) in place of 26 item occurrences in the 10 transactions. Node BH:8 is the shared prefix of 8 transactions; the TD:2 branch on the right holds the two transactions without BH (GD6 and GD8). Figure 4.4 draws this tree with the links of item XM.



Đường nét đứt: liên kết các nút cùng mục XM (từ bảng tiêu đề)

Bảng tiêu đề: BH 8 TD 6 VM 6 DM 3 XM 3

Hình 4.4. FP-tree của 10 giao dịch Phú Sa: mỗi nút ghi mục và số đếm; nét đứt nối các nút của mục XM

Figure 4.4. FP-tree of the 10 Phú Sa transactions: each node shows the item and its count; the dashed arrows join the nodes of item XM

Giai đoạn 2: khai phá. Duyệt các mục từ cuối bảng tiêu đề (ít phổ biến nhất) ngược lên. Với mục i , lần theo liên kết đến mọi nút i ; mỗi nút cho một đường tiền tố đi từ gốc đến cha của nó, mang số đếm của nút i . Tập các đường này là cơ sở mẫu điều kiện (conditional pattern base) của i . Đếm mục trong cơ sở, bỏ mục dưới ngưỡng, ta có cây FP điều kiện; mọi tổ hợp mục trên cây này ghép với i cho mẫu phổ biến có i làm hậu tố, và việc lặp lại đệ quy với từng mục của cây điều kiện.

Stage 2: mining. Take the items from the end of the header table (least frequent) upwards. For item i , follow the links to every i -node; each node gives a prefix path from the root to its parent, carrying the count of the i -node. The set of these paths is the conditional pattern base of i . Counting the items in the base and dropping those under the threshold gives the conditional FP-tree; every combination of items on this tree, joined with i , gives a frequent pattern with i as suffix, and the step repeats recursively for each item of the conditional tree.

Bài mẫu 5: khai phá mục XM. XM có ba nút: dưới đường BH, VM, dưới BH trực tiếp và dưới đường BH, DM. Ba đường tiền tố là {BH}, {BH, VM}, {BH, DM}, mỗi đường đếm 1. Trong cơ sở này BH có 3, VM có 1, DM có 1; chỉ BH đạt ngưỡng 3. Cây điều kiện của XM chỉ còn nhánh BH:3, cho hai mẫu: {XM} với 3 và {BH, XM} với 3.

Worked example 5: mining item XM. XM has three nodes: under the path BH, VM, directly under BH, and under the path BH, DM. The three prefix paths are {BH}, {BH, VM}, {BH, DM}, each with count 1. In this base BH has 3, VM has 1 and DM has 1; only BH reaches the threshold of 3. The conditional tree of XM keeps only the branch BH:3 and gives two patterns: {XM} with 3 and {BH, XM} with 3.

```
GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
thu_tu = ["BH", "TD", "VM", "DM", "XM"]
MIN = 3
vt = thu_tu.index("XM")
co_so = [[x for x in thu_tu[:vt] if x in t] for t in GD if "XM" in t]
print("Cơ sở mẫu điều kiện của XM:", co_so)
dem = {}
for duong in co_so:
    for i in duong:
        dem[i] = dem.get(i, 0) + 1
print("Đếm trong cơ sở mẫu:", dem)
pho_bien = [i for i, c in dem.items() if c >= MIN]
print("Cây điều kiện của XM giữ:", pho_bien)
print("Mẫu chứa XM:", [("XM", sum(1 for t in GD if "XM" in t))]
      + [(i + " XM", dem[i]) for i in pho_bien])
```

Kết quả chạy / Output

```
Cơ sở mẫu điều kiện của XM: [['BH'], ['BH', 'VM'], ['BH', 'DM']]
Đếm trong cơ sở mẫu: {'BH': 3, 'VM': 1, 'DM': 1}
Cây điều kiện của XM giữ: ['BH']
Mẫu chứa XM: [('XM', 3), ('BH XM', 3)]
```

Lập lại cho các mục còn lại ta được bảng sau. Với DM, ba đường tiền tố {BH, VM}, {TD}, {BH} cho BH 2, VM 1, TD 1, không mục nào đạt 3, nên chỉ còn mẫu {DM}. Với VM, cây điều kiện có BH 5 và TD 4; khai phá tiếp trên cây này cho thêm {BH, VM}, {TD, VM} và {BH, TD, VM}.

Repeating for the other items gives the next table. For DM, the three prefix paths {BH, VM}, {TD}, {BH} give BH 2, VM 1, TD 1, none reaching 3, so only the pattern {DM} remains. For VM, the conditional tree has BH 5 and TD 4; mining that tree further adds {BH, VM}, {TD, VM} and {BH, TD, VM}.

Hậu tố <i>Suffix</i>	Cơ sở mẫu điều kiện <i>Conditional pattern base</i>	Đếm mục trong cơ sở <i>Item counts in the base</i>	Mẫu phổ biến (đếm) <i>Frequent patterns (count)</i>
XM	{BH}:1; {BH, VM}:1; {BH, DM}:1	BH 3, VM 1, DM 1 ; giữ BH	XM (3); BH XM (3)
DM	{BH, VM}:1; {TD}:1; {BH}:1	BH 2, TD 1, VM 1 ; không giữ mục nào	DM (3)
VM	{BH, TD}:3; {BH}:2; {TD}:1	BH 5, TD 4 ; giữ BH, TD	VM (6); BH VM (5); TD VM (4); BH TD VM (3)
TD	{BH}:4	BH 4 ; giữ BH	TD (6); BH TD (4)
BH	rỗng <i>empty</i>	—	BH (8)

```

GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
thu_tu = ["BH", "TD", "VM", "DM", "XM"]
MIN = 3

def khai_thac(co_so, hau_to, ra):
    cnt = {}
    for duong, c in co_so:
        for i in duong:
            cnt[i] = cnt.get(i, 0) + c
    for i, c in cnt.items():
        if c < MIN:
            continue
        mau = (i,) + hau_to
        ra[mau] = c
        moi = [(d[:d.index(i)], w) for d, w in co_so if i in d]
        khai_thac(moi, mau, ra)

goc = [(x for x in thu_tu if x in t), 1) for t in GD]
ra = {}
khai_thac(goc, (), ra)
print(len(ra), "mẫu phổ biến")
for m, c in sorted(ra.items(), key=lambda x: (len(x[0]), -x[1])):
    print(" ".join(m), c)

```

Kết quả chạy / Output

```

10 mẫu phổ biến
BH 8
TD 6
VM 6
DM 3
XM 3
BH VM 5
BH TD 4
TD VM 4
BH XM 3
BH TD VM 3

```

Hàm đệ quy trên cho đúng 10 mẫu phổ biến, trùng 10 tập của Apriori (5 tập một mục, 4 tập hai mục, 1 tập ba mục); chỉ khác thứ tự liệt kê mục trong mẫu, vì FP-growth viết theo thứ tự tần số giảm (BH, TD, VM, DM, XM). Đoạn mã chỉ khai thác trên danh sách đường tiền tố kèm số đếm, không dựng cây, nhưng cùng nguyên lý cơ sở mẫu điều kiện.

The recursive function above returns exactly 10 frequent patterns, the same as the 10 sets of Apriori (5 single-item, 4 two-item, 1 three-item); only the order of items inside a pattern differs, because FP-growth writes them in descending frequency (BH, TD, VM, DM, XM). The code mines a list of prefix paths with counts rather than building the tree, but follows the same principle of the conditional pattern base.

PHẢI THUỘC / MEMORISE — FP-growth: (1) quét lần một, đếm và sắp mục theo tần số giảm; (2) quét lần hai, chèn giao dịch đã sắp vào FP-tree; (3) từ mục ít phổ biến nhất, lấy cơ sở mẫu điều kiện, lọc theo ngưỡng, khai phá đệ quy. Không sinh ứng viên.

FP-growth: (1) first scan, count and sort items by descending frequency; (2) second scan, insert the sorted transactions into the FP-tree; (3) from the least frequent item, take the conditional pattern base, filter by the threshold and mine recursively. No candidates are generated.

LỖI HAY GẶP / COMMON MISTAKE — Chèn giao dịch vào cây mà chưa sắp theo thứ tự tần số toàn cục. Nếu mỗi giao dịch sắp theo một thứ tự riêng, các giao dịch không chung

tiền tố, cây phình to và cơ sở mẫu điều kiện sai. Cả 10 giao dịch phải dùng cùng một thứ tự BH, TD, VM, DM, XM.

Inserting transactions into the tree before sorting them in the global frequency order. If each transaction is sorted in its own order, transactions do not share prefixes, the tree swells and the conditional pattern bases are wrong. All 10 transactions must use the same order BH, TD, VM, DM, XM.

LỖI HAY GẶP / COMMON MISTAKE — Quên lọc theo ngưỡng trong cơ sở mẫu điều kiện. Cơ sở của XM có VM đếm 1 và DM đếm 1; nếu giữ lại, ta sinh mẫu {VM, XM} và {DM, XM} chỉ xuất hiện 1 lần, dưới ngưỡng 3. Đếm trong cơ sở là số đếm của hậu tố cùng với mục đó, cộng theo số đếm của từng đường.

Forgetting to filter by the threshold inside the conditional pattern base. The base of XM has VM with count 1 and DM with count 1; if kept, the patterns {VM, XM} and {DM, XM} would be produced although they occur once, under the threshold of 3. The count in the base is the count of the suffix together with that item, summed over the count of each path.

4.9. So sánh Apriori và FP-growth; luật nhiều mức và luật định lượng — 4.9. Apriori compared with FP-growth; multilevel and quantitative rules

Cả hai thuật toán đều chính xác: với cùng minsup chúng trả về cùng tập phổ biến, nên sự khác biệt nằm ở chi phí. Bảng sau so sánh theo các tiêu chí thường gặp trong câu hỏi lý thuyết, kèm số đo trên dữ liệu Phú Sa.

Both algorithms are exact: with the same minsup they return the same frequent itemsets, so the difference lies in cost. The next table compares them on the criteria that usually appear in theory questions, with measurements on the Phú Sa data.

Tiêu chí Criterion	Apriori	FP-growth
Ý tưởng Idea	Sinh ứng viên theo mức, tia bằng tính chất đóng xuống, đếm <i>Generate candidates level by level, prune by downward closure, count</i>	Nén dữ liệu vào FP-tree, khai phá đệ quy trên cơ sở mẫu điều kiện <i>Compress the data into an FP-tree and mine recursively on conditional pattern bases</i>
Ứng viên Candidates	Sinh và đếm nhiều ứng viên; ở dữ liệu Phú Sa đếm 16 tập <i>Generates and counts many candidates; on the Phú Sa data 16 sets are counted</i>	Không sinh ứng viên <i>Generates no candidates</i>
Quét cơ sở dữ liệu Database scans	Một lần mỗi mức: 3 lần ở dữ liệu Phú Sa <i>One per level: 3 on the Phú Sa data</i>	Hai lần: một để đếm mục, một để dựng cây <i>Two: one to count items, one to build the tree</i>
Bộ nhớ Memory	Giữ danh sách ứng viên; bùng nổ khi minsup thấp <i>Holds the candidate list; explodes when minsup is low</i>	Giữ cây; nhỏ khi giao dịch chung nhiều tiền tố (12 nút ở dữ liệu Phú Sa) <i>Holds the tree; small when transactions share many prefixes (12 nodes on the Phú Sa data)</i>

Tiêu chí <i>Criterion</i>	Apriori	FP-growth
Khi nào hợp <i>When it suits</i>	Dữ liệu nhỏ, cần dễ cài đặt, minsup cao <i>Small data, ease of implementation, high minsup</i>	Dữ liệu lớn, đặc, minsup thấp <i>Large, dense data and low minsup</i>
Kết quả <i>Result</i>	Cùng tập phổ biến (10 tập) <i>The same frequent itemsets (10 sets)</i>	Cùng tập phổ biến (10 tập) <i>The same frequent itemsets (10 sets)</i>

```

from itertools import combinations
GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
ORD = "TD VM DM XM BH".split()
MIN = 3
C, k, quet, dem_uv, so_pb = [(i,) for i in ORD], 1, 0, 0, 0
while C:
    quet += 1
    dem_uv += len(C)
    L = [c for c in C if sum(1 for t in GD if set(c) <= t) >= MIN]
    so_pb += len(L)
    C = []
    for a, b in combinations(L, 2):
        u = tuple(sorted(set(a) | set(b), key=ORD.index))
        if (len(u) == k + 1 and u not in C
            and all(s in L for s in combinations(u, k))):
            C.append(u)
    k += 1
print("Apriori: quét", quet, "lần, đếm", dem_uv, "ứng viên,",
      so_pb, "tập phổ biến")
print("Số tập mục có thể có của 5 sản phẩm:", 2 ** 5 - 1)

```

Kết quả chạy / Output

```

Apriori: quét 3 lần, đếm 16 ứng viên, 10 tập phổ biến
Số tập mục có thể có của 5 sản phẩm: 31

```

FP-growth không phải lúc nào cũng nhanh hơn. Nếu giao dịch thưa và ít chung tiền tố, cây gần bằng số mục của cơ sở dữ liệu và việc dựng cây tốn công. Một hướng thứ ba là Eclat, lưu mỗi mục kèm danh sách mã giao dịch chứa nó (dạng dọc) và đếm tập lớn bằng cách giao hai danh sách.

FP-growth is not always faster. If transactions are sparse and share few prefixes, the tree approaches the size of the database and building it is costly. A third approach is Eclat, which stores each item with the list of transaction codes that contain it (vertical layout) and counts a larger set by intersecting two lists.

Luật nhiều mức (multilevel association rules) dùng cây phân cấp khái niệm: DM và XM thuộc nhóm "trả góp" (TG). Ở mức lá, độ hỗ trợ của sản phẩm rời rạc thấp; lên mức nhóm, độ hỗ trợ cao hơn nên lộ ra luật mà mức lá bỏ sót. Mã sau thêm mục TG vào giao dịch có DM hoặc XM.

Multilevel association rules use a concept hierarchy: DM and XM belong to the group "instalment" (TG). At the leaf level the support of separate products is low; at the group level support is higher, which reveals rules the leaf level misses. The code that follows adds the item TG to every transaction holding DM or XM.

```

GD = [set(s.split()) for s in
      ("TD VM BH,TD BH,VM DM BH,TD VM BH,XM BH,TD DM,VM XM BH,"
       "TD VM,DM XM BH,TD VM BH").split(",")]
n = len(GD)
# mức cao: TG = trà góp (điện máy hoặc xe máy)
GD2 = [t | {"TG"} if t & {"DM", "XM"} else t for t in GD]

def dem(tap, co_so):
    return sum(1 for t in co_so if tap <= t)

for ten in ("DM", "XM", "TG"):
    print(ten, dem({ten}, GD2), f"{dem({ten}, GD2) / n:.0%}")
n_ab, n_a = dem({"TG", "BH"}, GD2), dem({"TG"}, GD2)
conf = n_ab / n_a
print(f"TG => BH: sup {n_ab / n:.0%}, conf {conf:.0%},"
      f" lift {conf / (dem({'BH'}, GD2) / n):.2f}")

```

Kết quả chạy / Output

```

DM 3 30%
XM 3 30%
TG 5 50%
TG => BH: sup 40%, conf 80%, lift 1.00

```

DM và XM chỉ đạt 30% mỗi sản phẩm, sát ngưỡng, còn nhóm TG đạt 50%. Luật TG $\Rightarrow$ BH có support 40%, confidence 80% nhưng lift 1,00, tức độc lập với bảo hiểm: luật mức cao đạt ngưỡng mà không có giá trị gợi ý. Thực hành thường đặt minsup thấp hơn ở mức sâu hơn, và loại luật mức thấp khi nó chỉ lặp lại luật của mức cha với cùng độ tin cậy.

DM and XM reach only 30% each, close to the threshold, while the group TG reaches 50%. Rule TG $\Rightarrow$ BH has support 40%, confidence 80% but lift 1.00, i.e. it is independent of insurance: the high-level rule meets the thresholds without any suggestion value. In practice minsup is set lower at deeper levels, and a low-level rule is dropped when it merely repeats its parent rule at the same confidence.

Luật định lượng (quantitative association rules) có thuộc tính số như thu nhập hoặc dư nợ. Cách làm là rời rạc hóa thuộc tính thành khoảng (Chương 3), coi mỗi khoảng là một mục, rồi áp Apriori: ví dụ mục "thu nhập từ 10 đến dưới 20 triệu đồng". Chọn khoảng quá hẹp thì support thấp, quá rộng thì luật mất ý nghĩa, nên các điểm cắt thường thử nhiều phương án. Luật nhiều chiều dùng nhiều thuộc tính của khách (tuổi, nghề, sản phẩm) trong cùng một luật.

Quantitative association rules involve numeric attributes such as income or outstanding balance. The method is to discretise the attribute into intervals (Chapter 3), treat each interval as an item and then apply Apriori: for example the item "income from 10 up to 20 million dong". Intervals that are too narrow give low support and ones that are too wide drain the rule of meaning, so several cut points are usually tried. Multidimensional rules use several customer attributes (age, occupation, product) in one rule.

LỖI HAY GẶP / COMMON MISTAKE — Cho rằng FP-growth cho kết quả khác Apriori. Hai thuật toán cho cùng tập phổ biến; nếu hai kết quả lệch nhau thì một bản cài đặt có lỗi (thường là sai thứ tự sắp mục hoặc quên đếm giao dịch có số đếm lớn hơn 1 trong cơ sở mẫu điều kiện).

Believing that FP-growth gives a different result from Apriori. The two return the same frequent

itemsets; if the results differ, one implementation is faulty (typically a wrong item order, or forgetting to weight a path by its count in the conditional pattern base).

4.10. Mẫu tuần tự, ứng dụng đề xuất sản phẩm trên ứng dụng Phú Sa và giới hạn — 4.10. Sequential patterns, product recommendation in the Phú Sa app and limitations

Luật kết hợp coi giao dịch là tập, không có thứ tự. Khi dữ liệu có thời gian, thứ tự mang thông tin: khách vay tiền mặt trước rồi mới mở thẻ khác với khách mở thẻ trước. Khai phá mẫu tuần tự (sequential pattern mining) tìm các chuỗi con xuất hiện theo thứ tự trong nhiều chuỗi khách. Chuỗi $\langle VM, BH \rangle$ nghĩa là VM đến trước BH, không cần liên kề; độ hỗ trợ là số khách có chuỗi con đó chia cho tổng số khách. Hai thuật toán quen thuộc là GSP, theo kiểu Apriori, và PrefixSpan, theo kiểu tăng trưởng mẫu như FP-growth.

Association rules treat a transaction as a set with no order. When the data carry time, order carries information: a customer who takes a cash loan first and opens a card later differs from one who opens the card first. Sequential pattern mining finds subsequences that occur in order in many customer sequences. The sequence $\langle VM, BH \rangle$ means VM comes before BH, not necessarily next to it; support is the number of customers holding that subsequence divided by the total number of customers. Two familiar algorithms are GSP, in the Apriori style, and PrefixSpan, in the pattern-growth style of FP-growth.

Dữ kiện đề bài: chuỗi sản phẩm theo thời gian của năm khách, ngưỡng hỗ trợ 40% (2 trên 5 khách).

Exercise data: the time-ordered product sequences of five customers, with a support threshold of 40% (2 of 5 customers).

Khách Customer	Chuỗi sản phẩm theo thời gian Product sequence in time order
KH01	$\langle VM, BH, TD \rangle$
KH02	$\langle VM, TD, BH \rangle$
KH03	$\langle XM, BH, TD \rangle$
KH04	$\langle VM, BH \rangle$
KH05	$\langle DM, VM, BH \rangle$

```
CHUOI = [[ "VM", "BH", "TD"], [ "VM", "TD", "BH"], [ "XM", "BH", "TD"],
          [ "VM", "BH"], [ "DM", "VM", "BH"]]
MIN = 2 # 40% của 5 khách

def chua(chuoi, a, b):
    if a not in chuoi:
        return False
    return b in chuoi[chuoi.index(a) + 1:]

cap = {}
for x in "TD VM DM XM BH".split():
    for y in "TD VM DM XM BH".split():
        if x != y:
            d = sum(1 for s in CHUOI if chua(s, x, y))
```

```

if d >= MIN:
    cap[(x, y)] = d
for (x, y), d in sorted(cap.items(), key=lambda t: -t[1]):
    print(f"<{x}, {y}> {d}/{len(CHUOI)} = {d / len(CHUOI):.0%}")

```

Kết quả chạy / Output

```

<VM, BH> 4/5 = 80%
<VM, TD> 2/5 = 40%
<BH, TD> 2/5 = 40%

```

Ba chuỗi hai phần tử đạt ngưỡng: $\langle VM, BH \rangle$ có ở 4 khách (KH01, KH02, KH04, KH05), $\langle VM, TD \rangle$ ở KH01 và KH02, $\langle BH, TD \rangle$ ở KH01 và KH03. Chuỗi ngược $\langle TD, BH \rangle$ chỉ có ở KH02, dưới ngưỡng. Hai chuỗi $\langle BH, TD \rangle$ và $\langle TD, BH \rangle$ là hai mẫu khác nhau, điều mà tập mục không phân biệt được.

Three two-element sequences reach the threshold: $\langle VM, BH \rangle$ is held by 4 customers (KH01, KH02, KH04, KH05), $\langle VM, TD \rangle$ by KH01 and KH02, $\langle BH, TD \rangle$ by KH01 and KH03. The reverse $\langle TD, BH \rangle$ occurs only for KH02, under the threshold. $\langle BH, TD \rangle$ and $\langle TD, BH \rangle$ are different patterns, which itemsets cannot tell apart.

Ứng dụng di động Phú Sa dùng luật để đề xuất sản phẩm cho khách. Quy trình năm bước: (1) khai phá luật trên giao dịch lịch sử; (2) giữ luật qua ba ngưỡng support, confidence và lift lớn hơn 1; (3) với mỗi khách, chọn luật có vế trái nằm trong sản phẩm khách đang có và vế phải là sản phẩm khách chưa có; (4) xếp theo lift hoặc confidence và hiển thị một đến hai đề xuất; (5) đo tỷ lệ khách chấp nhận bằng thử nghiệm đối chứng. Bảng sau áp dụng quy trình này cho bốn luật mạnh của chương.

The Phú Sa mobile app uses rules to recommend products to customers. The process has five steps: (1) mine rules from historical transactions; (2) keep rules that pass three thresholds, support, confidence and lift above 1; (3) for each customer, pick rules whose left side is within the products the customer holds and whose right side is a product the customer lacks; (4) rank by lift or confidence and show one or two recommendations; (5) measure the acceptance rate in a controlled test. The next table applies this process to the four strong rules of the chapter.

Luật Rule	Lift	Quyết định Decision	Lý do Reason
TD, BH $\Rightarrow$ VM	1,25	Đề xuất vay tiền mặt cho khách có thẻ và bảo hiểm <i>Recommend a cash loan to customers with the card and insurance</i>	Lift cao nhất, support 0,3 <i>Highest lift, support 0.3</i>
XM $\Rightarrow$ BH	1,25	Đề xuất sau khi kiểm thêm dữ liệu <i>Recommend after checking more data</i>	Chỉ 3 giao dịch; cần mẫu lớn hơn <i>Only 3 transactions; a larger sample is needed</i>
VM $\Rightarrow$ BH	1,0417	Không lập chiến dịch riêng <i>No separate campaign</i>	Gần độc lập; tần suất nền 80% đã cao <i>Close to independence; the base rate of 80% is already high</i>

Luật <i>Rule</i>	Lift	Quyết định <i>Decision</i>	Lý do <i>Reason</i>
TD, VM $\Rightarrow$ BH	0,9375	Không đề xuất <i>Do not recommend</i>	Tương quan âm <i>Negative correlation</i>

Cần hỏi bộ phận kinh doanh xem bảo hiểm khoản vay có được bán kèm theo quy trình hay không. Nếu có, tần suất nền 80% phản ánh quy trình bán chứ không phải nhu cầu của khách, và mọi luật có BH ở vế phải cần đọc theo hướng đó.

The sales department should be asked whether loan insurance is bundled by the sales process. If it is, the base rate of 80% reflects the process and not customer demand, and every rule with BH on the right side must be read in that light.

Giới hạn của luật kết hợp: (1) tương quan không phải nhân quả; (2) 10 giao dịch quá ít, một giao dịch đổi chiều làm support lệch 10 điểm phần trăm; (3) sản phẩm phổ biến như BH chiếm nhiều luật mà ít giá trị; (4) luật phản ánh quá khứ, cần khai phá lại định kỳ khi sản phẩm thay đổi; (5) luật gắn với dữ liệu cá nhân của khách, nên việc dùng phải có cơ sở pháp lý và mục đích rõ ràng (Chương 9).

Limitations of association rules: (1) correlation is not causation; (2) 10 transactions are too few, and one transaction changing side moves support by 10 percentage points; (3) popular products such as BH fill many rules of little value; (4) rules reflect the past and must be mined again regularly as products change; (5) rules are tied to customers' personal data, so their use needs a legal basis and a clear purpose (Chapter 9).

PHẢI THUỘC / MEMORISE — Tương quan không phải nhân quả: luật cho biết hai sản phẩm hay đi cùng nhau, không cho biết mua sản phẩm này khiến khách mua sản phẩm kia. Chỉ thử nghiệm đối chứng mới đo được tác động của một lời đề xuất.

Correlation is not causation: a rule says two products go together, not that buying one makes the customer buy the other. Only a controlled test can measure the effect of a recommendation.

LỖI HAY GẶP / COMMON MISTAKE — Triển khai luật từ mẫu nhỏ mà không kiểm lại. Luật XM $\Rightarrow$ BH với confidence 100% chỉ dựa trên 3 giao dịch; thêm một giao dịch có XM mà không có BH, confidence tụt còn $3/4 = 75\%$. Phải kiểm trên dữ liệu mới trước khi dùng.

Deploying rules from a small sample without re-checking. Rule XM $\Rightarrow$ BH with 100% confidence rests on 3 transactions; one more transaction with XM but without BH drops the confidence to $3/4 = 75\%$. Check on new data before use.

4.11. Bảng thuật ngữ — 4.11. Glossary

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Luật kết hợp	association rule	Quy luật $X \Rightarrow Y$: giao dịch có X thì thường có Y <i>A rule $X \Rightarrow Y$: transactions with X tend to have Y</i>
Phân tích giỏ hàng	market basket analysis	Tìm các mặt hàng hay nằm cùng giỏ <i>Finding articles that often sit in the same basket</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Bán chéo	cross-selling	Đề xuất sản phẩm thứ hai cho khách đã có sản phẩm thứ nhất <i>Offering a second product to a customer who holds the first</i>
Giao dịch	transaction	Tập mục xuất hiện cùng nhau trong một lần <i>The set of items occurring together in one event</i>
Tập mục	itemset	Tập hợp một hay nhiều mục; k mục gọi là k-tập mục <i>A set of one or more items; k items form a k-itemset</i>
Số đếm hỗ trợ	support count	Số giao dịch chứa tập mục <i>The number of transactions containing the itemset</i>
Độ hỗ trợ	support	Số đếm chia cho tổng số giao dịch <i>The count divided by the total number of transactions</i>
Tập phổ biến	frequent itemset	Tập mục có độ hỗ trợ không nhỏ hơn minsup <i>An itemset whose support is at least minsup</i>
Ngưỡng hỗ trợ tối thiểu	minimum support (minsup)	Ngưỡng để một tập mục được coi là phổ biến <i>The threshold for an itemset to count as frequent</i>
Độ tin cậy	confidence	Xác suất có Y khi đã có X <i>The probability of Y given X</i>
Ngưỡng tin cậy tối thiểu	minimum confidence (minconf)	Ngưỡng để một luật được giữ <i>The threshold for a rule to be kept</i>
Luật mạnh	strong rule	Luật đạt cả minsup và minconf <i>A rule that meets both minsup and minconf</i>
Độ nâng	lift	Confidence chia cho support của vế phải; 1 là độc lập <i>Confidence divided by the support of the consequent; 1 means independence</i>
Leverage	leverage	Hiệu giữa tần suất chung quan sát và kỳ vọng <i>Observed minus expected joint frequency</i>
Conviction	conviction	Đo mức luật sai ít hơn so với khi độc lập <i>How much less often the rule fails than under independence</i>
Tính chất đóng xuống	downward closure (Apriori property)	Tập con của tập phổ biến đều phổ biến <i>Every subset of a frequent itemset is frequent</i>
Tập ứng viên	candidate itemset	Tập mục được sinh ra để đếm <i>An itemset generated to be counted</i>

Thuật ngữ <i>Term (Vietnamese)</i>	Tiếng Anh <i>English</i>	Giải thích ngắn <i>Short definition</i>
Ghép	join	Hợp hai tập cỡ $k-1$ thành tập cỡ k <i>Uniting two sets of size $k-1$ into a set of size k</i>
Tia	prune	Loại ứng viên có tập con không phổ biến <i>Removing a candidate that has an infrequent subset</i>
Tập phổ biến đóng	closed frequent itemset	Phổ biến, không có tập cha cùng số đếm <i>Frequent, with no proper superset of the same count</i>
Tập phổ biến cực đại	maximal frequent itemset	Phổ biến, không có tập cha phổ biến <i>Frequent, with no frequent proper superset</i>
Cây FP	FP-tree	Cây tiền tố nén các giao dịch đã sắp theo tần số <i>A prefix tree compressing transactions sorted by frequency</i>
Cơ sở mẫu điều kiện	conditional pattern base	Các đường tiền tố dẫn tới một mục, kèm số đếm <i>The prefix paths leading to an item, with counts</i>
Mẫu tuần tự	sequential pattern	Chuỗi con xuất hiện theo thứ tự trong nhiều chuỗi <i>A subsequence occurring in order in many sequences</i>

4.12. Bài tập vận dụng có lời giải — *Applied exercises with solutions*

Bài 1. Apriori và luật kết hợp trên sáu giao dịch — *Exercise 1. Apriori and association rules on six transactions*

Công ty Phú Sa ghi sáu giao dịch đăng ký sản phẩm trên ứng dụng (dữ kiện đề bài): $G1 = \{TD, BH\}$; $G2 = \{TD, VM, BH\}$; $G3 = \{VM, DM\}$; $G4 = \{TD, VM, BH\}$; $G5 = \{TD, VM\}$; $G6 = \{VM, BH, DM\}$. Cho $\text{minsup} = 50\%$ và $\text{minconf} = 75\%$. (a) Tìm mọi tập phổ biến bằng Apriori. (b) Sinh các luật mạnh. (c) Tính lift của từng luật mạnh và nêu luật nào không nên dùng để đề xuất.

Phú Sa records six product-registration transactions on its app (exercise data): $G1 = \{TD, BH\}$; $G2 = \{TD, VM, BH\}$; $G3 = \{VM, DM\}$; $G4 = \{TD, VM, BH\}$; $G5 = \{TD, VM\}$; $G6 = \{VM, BH, DM\}$. Take $\text{minsup} = 50\%$ and $\text{minconf} = 75\%$. (a) Find all frequent itemsets with Apriori. (b) Generate the strong rules. (c) Compute the lift of each strong rule and say which rule should not be used for recommendation.

1. Ngưỡng đếm: $50\% * 6 = 3$ giao dịch. C1 và số đếm: TD 4, BH 4, VM 5, DM 2. DM có $2 < 3$ nên bị loại; $L1 = \{TD\}, \{BH\}, \{VM\}$.

Count threshold: $50\% * 6 = 3$ transactions. C1 and counts: TD 4, BH 4, VM 5, DM 2. DM has $2 < 3$ and is dropped; $L1 = \{TD\}, \{BH\}, \{VM\}$.