化工与人工智能研究生课程，2024秋季

化工与人工智能

Chemical Engineering and Artificial Intelligence

霍　锋
刘亚伟

中国科学院过程工程研究所

第四章 数据预处理 (一)

清洗、集成、转化、归约

本章提纲

1. 数据清洗
2. 数据集成
3. 数据转化
4. 数据归约

什么是数据预处理

- **概念：**数据预处理是对原始数据进行清理和转换的过程，使其更适合于模型训练或数据分析。
- **原因：**数据预处理是数据挖掘和机器学习中非常关键的步骤，因为原始数据往往存在噪声、缺失值、不一致性等问题。有效的数据预处理可以提高模型的准确性和性能。
- **特点：**数据预处理的选择取决于具体的应用场景和数据特点。在预处理的过程中，仔细探索数据，了解数据特征，是进行有效预处理的前提。（学科的专业背景知识是必要的！）

数据预处理的主要任务

- **数据清洗：**处理缺失值、重复数据、异常值和不一致数据，以提高数据质量。
- **数据集成：**将来自不同数据源的数据整合成一个统一的数据集，以供后续分析。包括数据源合并、去重、消除冗余等。
- **数据变换：**将数据转换为合适的格式或结构。常见的变换操作包括归一化、标准化、离散化、数据编码（如将分类数据转换为数值）等，以便不同类型的数据可以被算法处理。
- **数据归约：**减少数据规模而保留重要信息，以提高处理效率。包括特征选择、特征提取、数据抽样、维度缩减（如主成分分析）等。

数据清洗

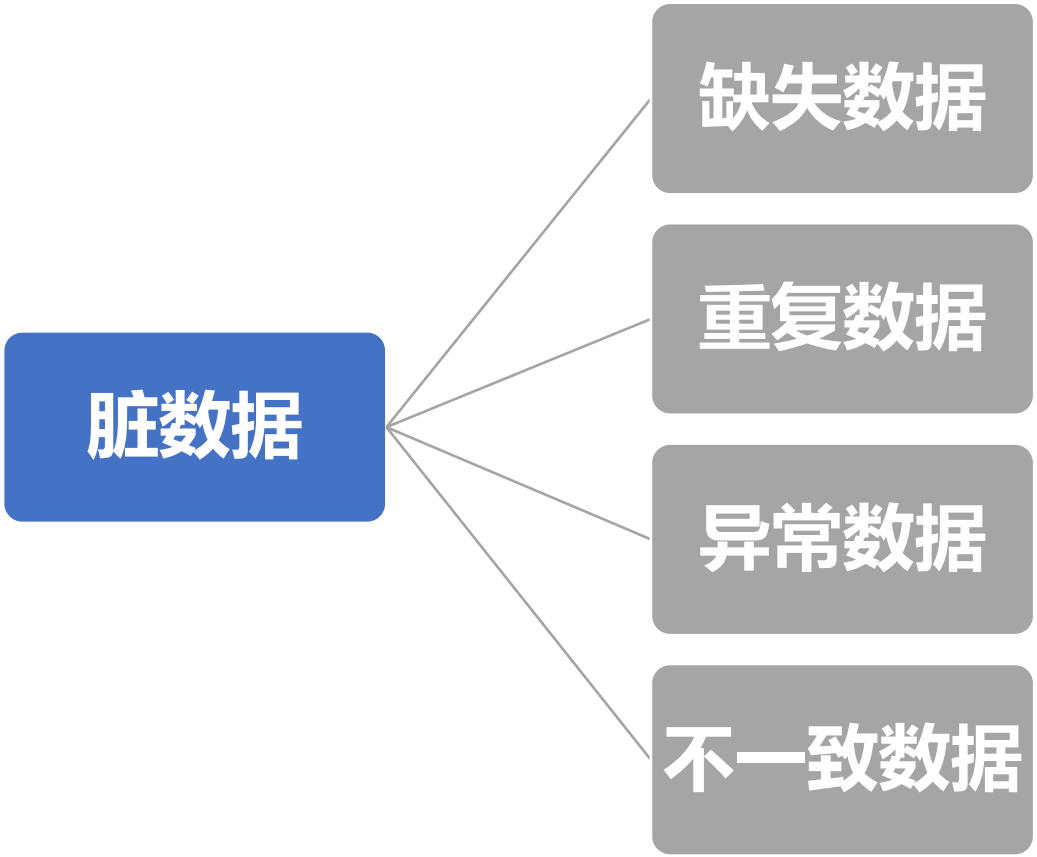
案例

背景：一个大型化工厂生产某种化学品。该厂每天都会收集关于生产过程的大量数据，如温度、压力、流量、原材料质量、产品质量等。这些数据对于生产过程的监控、优化和质量控制至关重要。然而，工厂最近注意到，一些设备故障和人为误操作可能导致数据记录中的一些异常值，需要对这些数据进行清洗。

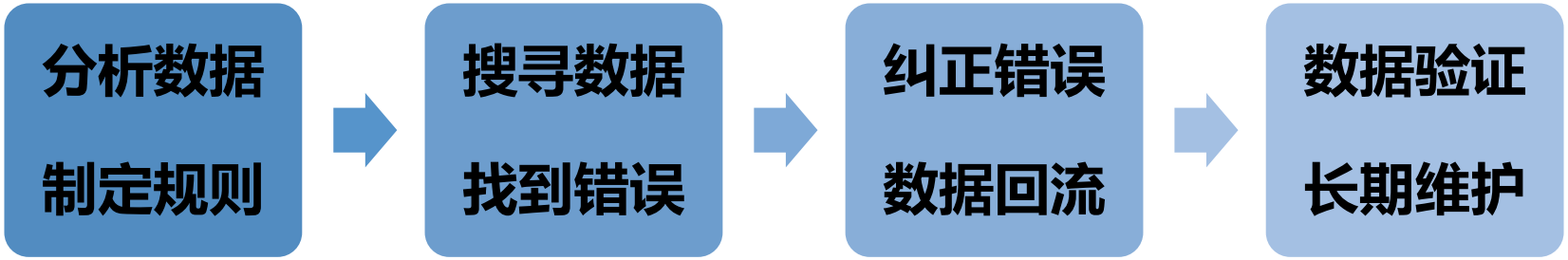
- **缺失值**：某些传感器在某些时段没有工作，导致数据记录中出现缺失值。
- **重复值**：设备故障导致一些值重复记录。
- **异常值**：由于设备故障或其他原因，某些传感器记录的值远超过正常范围，例如一个温度传感器突然记录了一个-200°C的值。
- **不一致值**：不同的设备可能使用不同的单位记录数据，例如一些设备记录压力为“帕斯卡”(Pa)，而其他设备记录为“大气压”(atm)。

什么是数据清洗

❖数据清洗：对“脏数据”进行处理



数据清洗整体步骤和难点



依赖于统计分析和可视化

难点：

- 数据清洗是一项十分繁重的工作，数据清洗在提高数据质量的同时要付出一定的代价，包括投入的时间、人力和物力成本
- 通常情况下，大数据集的数据清洗是一个系统性的工作，需要多方配合以及大量人员的参与，需要多种资源的支持。
- 数据清洗一般针对具体的应用，难以归纳统一方法和步骤。

数据清洗整体思路和策略

- **数据溯源：从源头找出问题**
  - **思路**：数据清洗的第一步是了解数据的来源和流转过程。通过数据溯源，找到“脏数据”产生的源头，分析问题出现的原因，如数据采集、传输、存储等环节可能导致的数据质量问题。
  - **策略**：在数据进入清洗流程前，尽可能了解数据的生成和传输路径，确定需要处理的异常、错误或噪声的性质。

- **数据审查：全面了解数据特征**
  - **思路**：在清洗数据之前，先审查数据，检查数据的基本特征，包括数据类型、分布情况、缺失值、异常值和重复值等。数据审查有助于确定数据中潜在的质量问题和清洗的重点。
  - **策略**：使用统计分析、数据可视化等手段来识别数据中的错误、异常值和不一致，生成初步的清洗规则。

数据清洗整体思路和策略

- **提取清洗规则和策略：制定清洗方案**
  - **思路**：根据数据审查结果，制定数据清洗的规则和策略。例如，如何处理缺失值、异常值、重复值、数据格式不一致等问题。这些规则应该根据数据业务背景、业务逻辑和需求来定制。
  - **策略**：使用行业标准、业务规则和统计方法来制定清洗规则。根据不同的清洗需求选择适当的策略，如删除、填补、格式转换等。

- **数据清洗：应用清洗规则**
  - **思路**：根据制定的清洗规则，对数据集中的“脏数据”进行处理，使其满足预期的要求。
  - **策略**：应用特定的清洗算法，按照规则对数据进行清洗。例如，对于缺失值，可以采用均值、中位数填充；对于异常值，可以进行剔除或替换；对于文本数据，可以进行规范化和编码。

数据清洗整体思路和策略

- **数据验证**：验证清洗效果
  - **思路**：清洗完成后，对数据进行验证，检查清洗后的数据质量，确保清洗策略有效。
  - **策略**：通过重新审查数据、生成统计报告、业务逻辑校验等方式来验证清洗效果。必要时，调整清洗策略并重新清洗。

- **数据监控与维护**：建立持续优化机制
  - **思路**：数据清洗不是一次性操作，应当持续监控数据质量，及时发现和处理新出现的“脏数据”。
  - **策略**：建立数据质量监控机制，定期或实时地检查数据，动态调整数据清洗策略，保证数据的长期质量。

数据清洗的一些规则

缺失数据

如果缺失值是**数值型**的，就根据该变量在其他所有对象的取值的平均值来填充该缺失的变量值；
如果缺失值是**非数值型**的，则使用众数来补齐该缺失的变量值。

异常数据

用统计分析的方法识别可能的错误值或异常值，如偏差分析、识别不遵守分布或回归方程的值，也可以用简单规则库(常识性规则、业务特定规则等）检查数据值

重复数据

合并或者清除是消重的基本方法。

不一致数据

定义完整性约束，也可通过分析数据发现联系。

缺失数据处理

姓名	身高	体重	…	血糖值
XX	XX	XX	…	XX
XX	XX	XX	…	XX
XX	XX		…	XX
XX	XX	XX	…	XX
XX	XX	XX	…	XX
XX	XX	XX	…	XX

缺失数据处理

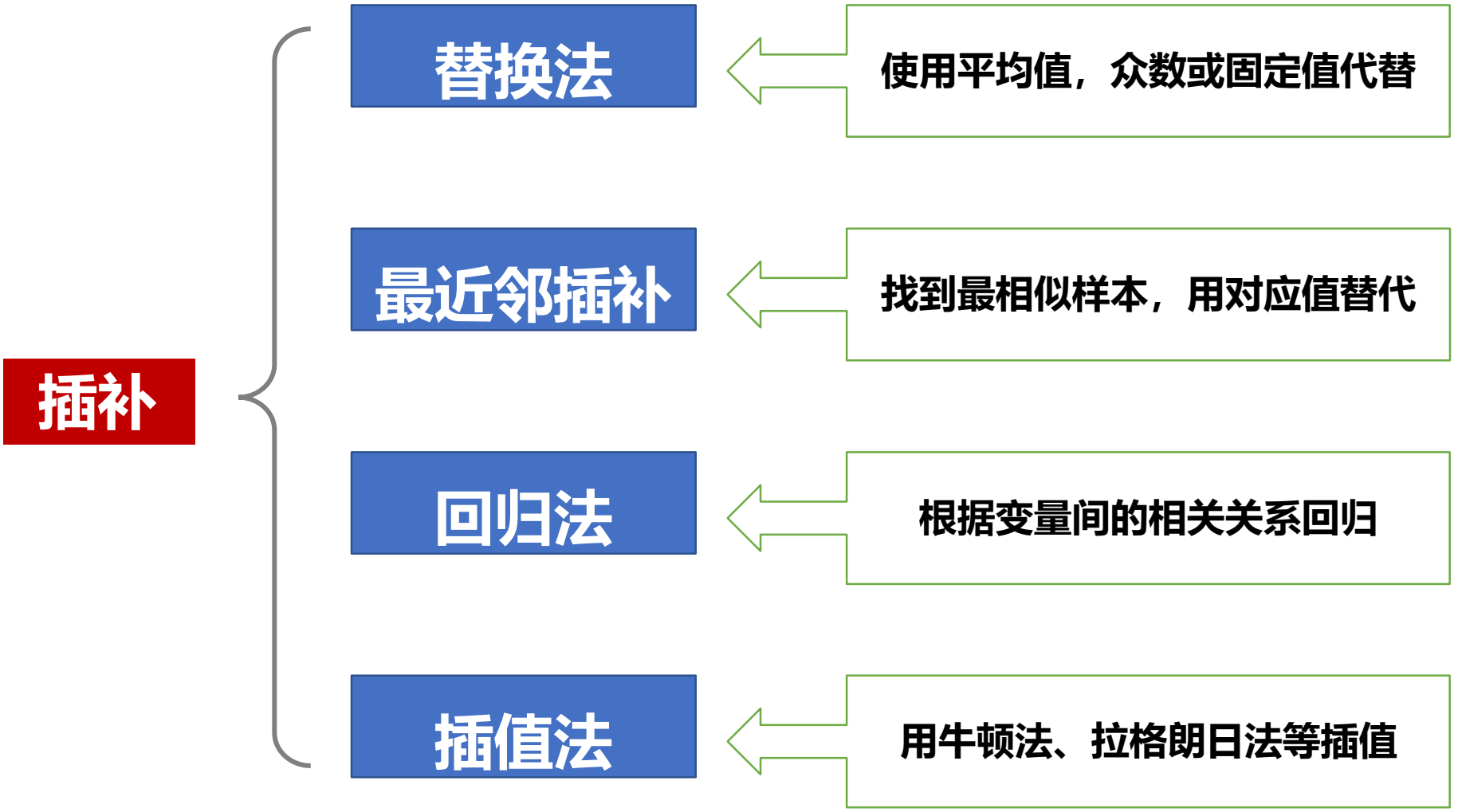
删去整条数据

优点：
简单，保证数据的准确性

缺点：
数据成本高，其余数据可能已包括大部分信息，删去浪费

插补

缺失数据处理

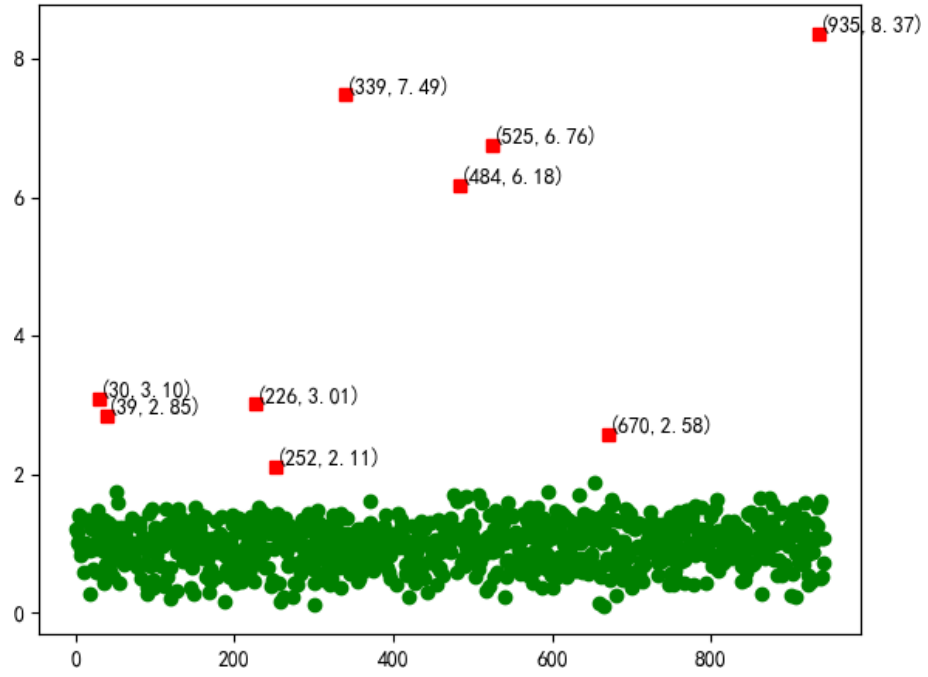

插补
替换法
使用平均值，众数或固定值代替
最近邻插补
找到最相似样本，用对应值替代
回归法
根据变量间的相关关系回归
插值法
用牛顿法、拉格朗日法等插值

重复数据

- **删除重复记录**：
  - **直接删除**：利用数据处理工具（如Pandas的drop_duplicates()方法），删除完全重复的行，保留一份完整的记录。
  - **条件删除**：如果记录部分重复，且其中一个字段（如时间戳、状态）有更高的可信度，可以设置条件删除，保留最可信的数据行。
- **合并重复数据**：
  - **数据合并**：当重复记录包含不同字段的信息时，可以将这些信息合并。例如，若两条记录的联系人地址不同，可以将它们合并为一条记录，并保留两个地址作为列表。

异常数据处理

- ❖ 删除异常值：直接删除明显的异常数据点（如明显错误的数
- ❖ 替换异常值：使用均值、中位数或其他合理的数值替换异常值。
- ❖ 异常值分析：检测数据是否有输入错误或者含有不合常理的数据



异常值检查

- 简单统计量分析
- 3σ准则
- 箱型图分析

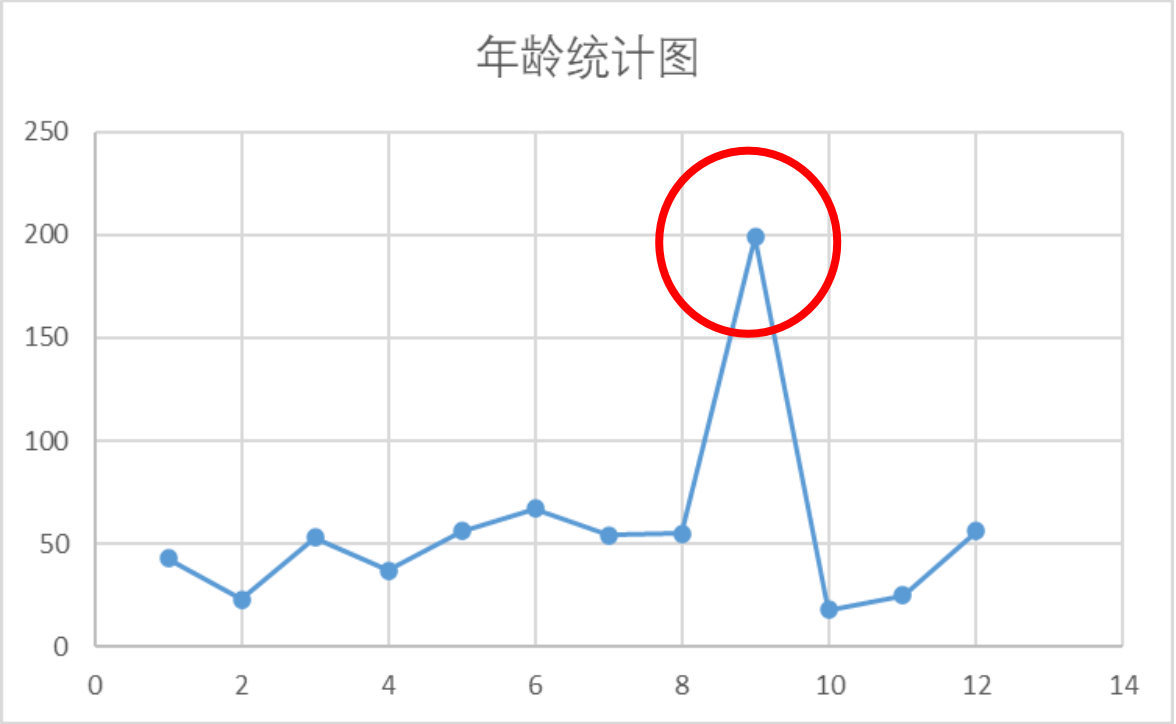
异常数据处理

简单统计量分析

计算数据的统计量，进行分析

最常用的是最大值最小值并分析是否合理

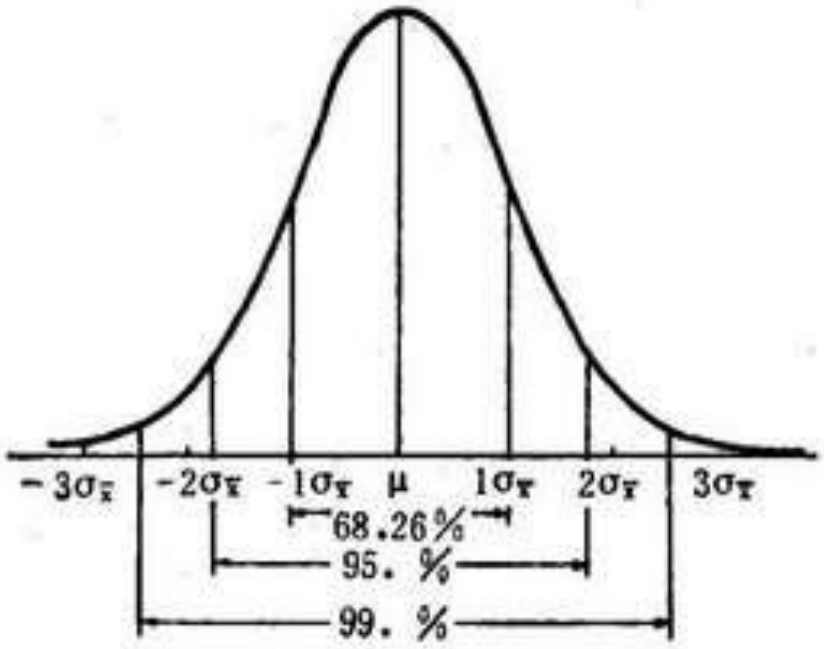
- 最大年龄是200?
- 体重出现负数?



异常数据处理

3σ准则	正态分布3倍标准差之外为异常

1倍标准差之内：68.3%
2倍标准差之内：95.5%
3倍标准差之内：99.7%



例子：
均值是10，标准差是2，定义远离3倍标准差就是异常值，在4~16之间的数字为正常

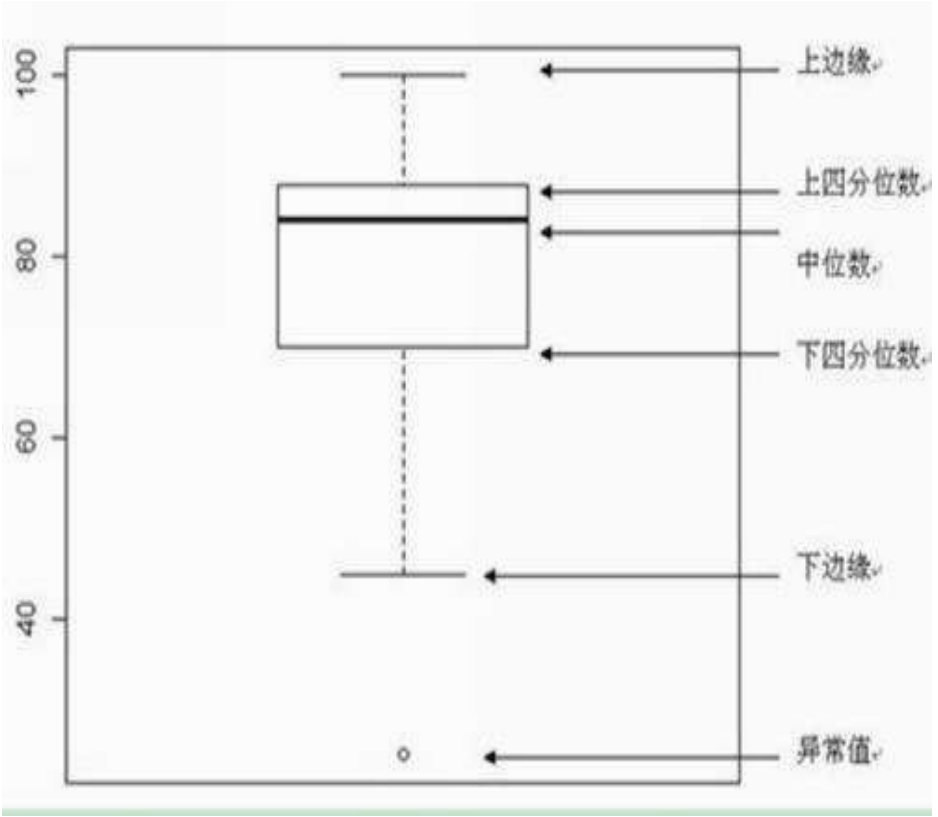
异常数据处理

箱型图分析 | 箱型图外部为异常

上四分位数：全部数据中有1/4的数据比它大，记作Q_U下四分位数：全部数据中有1/4的数据比它小，记作Q_L

Q_U~Q_L之间包含了一半的数据，记$IQR=Q_U-Q_L$

异常数据：小于$Q_L-1.5*IQR$，或者大于$Q_U+1.5*IQR$



箱型图具有比较好的鲁棒性

数据一致性分析

统一数据格式：

- **日期格式标准化**：将日期格式统一为标准形式，如YYYY-MM-DD，以确保数据处理的正确性。
- **大小写统一**：对文本数据进行大小写转换，如将所有国家名称转换为大写或小写，确保一致性。
- **单位转换**：对于数值数据，统一单位表示，例如将所有温度单位统一为摄氏度，或将货币单位统一为人民币。

查找和替换：

- **正则表达式**：使用正则表达式查找并替换文本中的不一致内容，例如，格式化电话号码、删除多余空格、统一缩写等。

数据一致性分析

数据映射：

- **类别映射**：将不同形式表示的同一类别映射为一致的值。例如，将“Male”、“M”映射为统一的“Male”。
- **错误纠正**：针对明显的拼写错误或误用的类别名进行纠正。

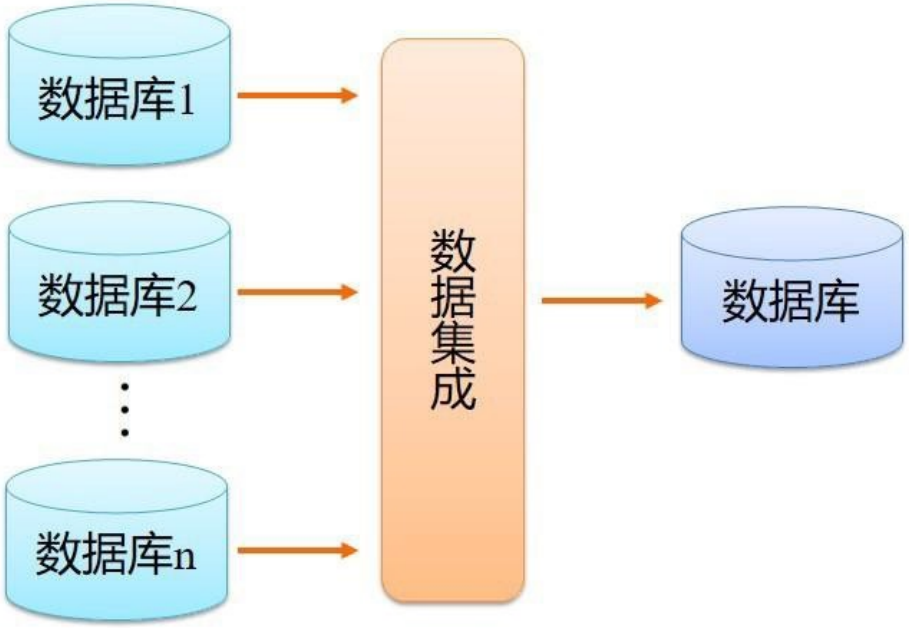
数据合并：

- **主键匹配**：当多源数据存在不一致时，可以基于主键（如ID）合并，确保同一对象在各个数据源中保持一致性。
- **多值整合**：对于同一对象的多值情况，可以根据优先级规则进行整合，保留最可信的信息。

数据集成

数据集成概念

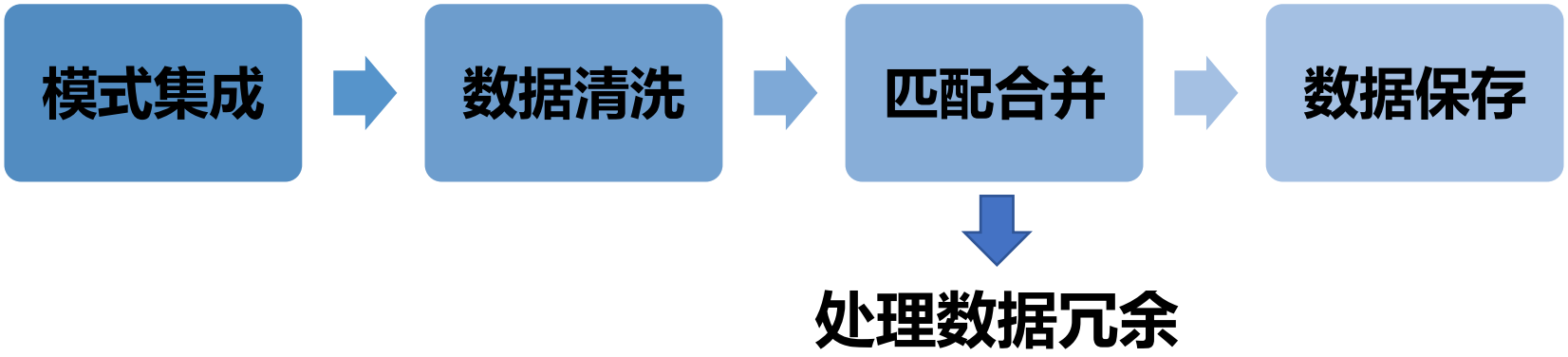
- **数据集成**是将来自不同来源的数据（如数据库、文件系统、应用程序、API、物联网设备等）进行**收集、清洗、转换和合并**，整合为一个统一的数据集或数据视图的过程。
- 数据集成的目标是**消除数据孤岛**，使各类数据在一个统一的平台中得到有效管理和利用，从而支持数据分析、业务决策、数据共享等需求。



数据集成特点

- **多源数据整合**：数据集成通常涉及多个数据源，这些数据源可能在结构、格式、存储方式上各不相同。数据集成通过统一数据格式和规范，将这些异构数据整合为一致的形式。
- **数据清洗与转换**：在数据集成过程中，通常需要对数据进行清洗（如去除噪声、处理缺失值、删除重复数据）和转换（如数据类型转换、格式转换、单位统一等），以确保数据的质量和一致性。
- **统一的数据视图**：数据集成最终形成一个统一的数据视图，使用户能够从中获取完整的、跨多个数据源的信息。统一的数据视图可以是一个数据仓库、数据湖、数据集市，或通过数据虚拟化实现。
- **支持数据分析和决策**：数据集成使企业或组织能够从不同来源的数据中获取综合的、完整的洞察，为数据分析、机器学习、业务决策提供支持。

数据集成流程



模式集成

- 确定需要集成的各个数据源，包括数据库、文件、云服务、API、传感器等。了解每个数据源的类型、结构、格式和内容，为后续集成工作做好准备。
- 对每个数据源进行分析，了解其数据格式、字段名称、数据类型、数据质量等。确定哪些数据需要集成，哪些数据可以被忽略，制定初步的数据集成策略。

数据库A

cust-id	name	height
1	丁兆云	1.68
2	张三	1.76

数据库B

cust-#	name	height
1	dzy	5.51
2	zs	5.77

数据集成

id	nameA	heightA	nameB	heightB
1	丁兆云	1.68	dzy	5.51
2	张三	1.76	zs	5.77

处理数据冗余

- 从多个数据源（可能是数据库、文件、API等）中收集、合并和提供统一视图的过程。在这个过程中，数据冗余是一个常见的问题。
- **数据冗余**是指存储了不必要的、重复的数据。如同一信息的多次存储、不同数据源之间的重叠、不一致的数据表示等。

数据库A

cust-id	name	3000m
1	丁兆云	13.24
2	张三	11.26

数据库B

cust-#	name	5000m
1	dzy	25.35
2	zs	21.27

数据集成

id	name	run
1	丁兆云	15.24
2	张三	12.14

- 可以通过人为识别找出冗余数据，或者通过**相关性分析**

相关性分析

- 用于检测和量化两个或多个变量之间的关联性。

- 如果两个属性高度相关，那么一个属性基本上可以用另一个属性来预测，这意味着其中一个属性可能是冗余的。例如，如果我们在气象数据集中有两个属性：摄氏度和华氏度的温度，那么这两个属性就是高度相关的，因为一个属性可以很容易地从另一个属性转换过来。

- 相关分析中常用的指标包括：
  - 皮尔逊相关系数：量化两个连续变量之间的线性关系。
  - 斯皮尔曼秩相关系数：非参数的方法，适用于连续或顺序分类变量。
  - 肯德尔的τ：另一种非参数的方法，适用于顺序分类变量。

皮尔逊相关系数

- 一种用于测量两个连续变量之间线性关系强度和方向的统计量。其值范围在-1和1之间。

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2(y_i - \bar{y})^2}}$$

  - **系数为正**：表示两个变量之间存在正相关关系。
  - **系数为负**：表示两个变量之间存在负相关关系。
  - **系数为0**：表示两个变量之间没有线性关系。

- 注意：
  - 皮尔逊相关系数只能捕捉到线性关系，对于非线性关系可能不是很有效。
  - 高皮尔逊相关系数并不意味着因果关系。例如，两个变量可能因为第三个未观察到的因素而相关，这称为混淆变量。如各个城市的病人与犯罪人数的关系（他们可能因为城市人口变多都在增加）。

斯皮尔曼秩相关系数

- 一种非参数统计方法，用于衡量两个变量的相关性，特别是当数据不满足正态分布或线性关系的假设时。它基于数据的秩（排名）而不是原始值来计算。不仅可以测量线性关系，还可以测量非线性关系。

$$r_s = 1 - \frac{6\sum d_i^2}{n(n^2-1)}$$

d_i两个变量的秩差　　n观测值的数量

- **系数为正**：表示两个变量之间存在正相关关系。
- **系数为负**：表示两个变量之间存在负相关关系。
- **系数为0**：表示两个变量之间没有线性关系。

案例：化肥生产

- 假设在一家化肥生产企业中，生产工程师研究了5批次的原料与其化肥效果。数据如下：

批次	原料质量等级	化肥效果等级
1	良	良
2	优	优
3	中	中
4	良	中
5	差	差

秩分配

- 优=4，良=3，中=2，差=1

批次	原料质量秩	化肥效果秩
1	3	3
2	4	4
3	2	2
4	3	2
5	1	1

计算秩差

批次	原料质量秩	化肥效果秩	秩差 d	d^2
1	3	3	0	0
2	4	4	0	0
3	2	2	0	0
4	3	2	1	1
5	1	1	0	0

计算斯皮尔曼秩相关系数

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

其中 n = 5 （观测值的数量）

$$r_s = 1 - \frac{6(1)}{5(24)}$$

$$r_s = 1 - \frac{6}{120}$$

$$r_s = 1 - 0.05$$

$$r_s = 0.95$$

表明原料质量等级与化肥效果等级之间存在非常强烈的正相关关系。这意味着原料质量的提高通常会导致化肥效果的提高。

肯德尔的τ

- 用于测量两个变量之间关系的非参数统计方法。它评估了两个变量之间的序列关系，即一个变量的值的增加（或减少）是否与另一个变量的值的增加（或减少）一致。肯德尔的τ主要用于测量等级数据的关联。
- 基于对数据对的配对比较来计算的。具体地说，它考虑了所有可能的数据对，并确定了多少对是“一致的”（即两个变量在两个观察值之间都增加或都减少）和多少对是“不一致的”（即一个变量增加而另一个变量减少）。

$$\tau = \frac{(n_c - n_d)}{n(n-1)/2}$$

n_c 一致对的数量， n_d不一致对的数量，
n 观察值数量

- τ = 1 表示完全一致的正相关。
- τ = -1 表示完全一致的负相关。
- τ = 0 表示没有相关性。

案例：催化剂

- 假设你是一个化学工程师，正在研究两种不同的催化剂A和B对某化学反应速率的影响。你对10个不同的反应条件进行了实验，并对这两种催化剂的效果进行了等级评定，从最低的1到最高的10。

实验	催化剂A的等级	催化剂B的等级
1	2	3
2	5	6
3	7	8
4	1	1
5	3	2
6	8	9
7	6	7
8	10	10
9	9	5
10	4	4

计算一致的对和计算不一致的对

- ✓ 实验1, 2
- ✓ 实验2, 7
- ✓ 实验3, 6
- ✓ 实验4, 5
- ✓ … (以此类推)

一致的对（A/B都变好或变差）数量 $n_c = 7$

- ✓ 实验2, 9
- ✓ 实验3, 9
- ✓ … (以此类推)
- ✓ 不一致的对（A/B变化不一致）数量 $n_d = 3$

$$\tau = \frac{(n_c - n_d)}{n(n-1)/2}$$
$$\tau = \frac{(7-3)}{10(10-1)/2}$$
$$\tau = \frac{4}{45}$$
$$\tau = 0.089$$

两种催化剂的效果具有轻微的正相关。

数据转化

数据转化

- **数据转化**是指将数据从一种格式或结构转换为另一种格式或结构，以便于后续分析和建模。
- **常见操作：**
  - **数值化**：将分类数据（如性别、颜色）转化为数值数据。例如，将“红色”、“蓝色”和“绿色”分别编码为 0、1 和 2。
  - **规范化**：使用Min-Max缩放或z-score标准化将特征的范围转化为[0, 1]或平均值为0、标准差为1。
  - **离散化**：将连续数据转化为离散类别。例如，将年龄数据分为“儿童”、“青年”、“中年”和“老年”等类别。
  - **特征构造**：从原始数据中生成新的特征或属性，如从日期特征中提取出“月份”或“星期几”。
  - **类型转换**：将数据从一种数据类型转换为另一种类型。例如，将字符串型日期 "2024-01-01" 转换为日期类型，或将浮点数转为整数类型。

数值化

- **数值化**是将非数值型数据（通常是分类数据或文本数据）转换为数值型数据的过程，以便用于统计分析或机器学习模型中。机器学习模型通常需要数值型输入，因此，将分类数据转化为数值数据是预处理的关键步骤。

案例 1：标签编码（Label Encoding）

标签编码是一种将分类数据转换为整数的数值化方法。每个类别会被映射为一个唯一的整数。

- **数据**：假设有一个关于“颜色”的分类特征，包含三个值：`['红色', '蓝色', '绿色']`。
- **数值化**：将每个颜色映射为一个整数：
  - 红色 → 0
  - 蓝色 → 1
  - 绿色 → 2

结果：`['红色', '蓝色', '绿色']` → `[0, 1, 2]`

数值化

❖ **数值化**是将非数值型数据（通常是分类数据或文本数据）转换为数值型数据的过程，以便用于统计分析或机器学习模型中。机器学习模型通常需要数值型输入，因此，将分类数据转化为数值数据是预处理的关键步骤。

案例 2：独热编码（One-Hot Encoding）

独热编码将每个类别转换为二进制向量。每个类别都由一个独立的特征列表示，如果某一行属于这个类别，则对应列的值为 1，否则为 0。

- **数据**：`['红色', '蓝色', '绿色']`
- **数值化**：
  - 红色 → `[1, 0, 0]`
  - 蓝色 → `[0, 1, 0]`
  - 绿色 → `[0, 0, 1]`

规范化： 归一化

- Min-Max缩放是一种**数据归一化技术**，用于将特征的值范围缩放到一个指定的范围，通常是[0, 1]。这种方法尤其适用于那些对输入特征的尺度或分布非常敏感算法，如梯度下降、神经网络等。
- Min-Max缩放的计算公式如下：

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad \text{or} \quad X_{norm} = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}}$$

- 这种缩放方法的一个优点是计算简单且直观。但它也有缺点，如受到**异常值的影响**。例如，如果一个特征中有一个非常大的异常值，其他的正常值在使用Min-Max缩放后可能会被压缩到一个非常小的范围内。
- 在使用Min-Max缩放之前，通常需要先处理异常值或考虑使用其他的缩放方法，如Z-score标准化。

案例：化工生产中的温度与压力数据归一化

- **背景**： 在一个化工生产过程中，工程师需要监控反应器的温度和压力。这两个参数对于保证生产过程的稳定性和产品的质量至关重要。假设工程师希望使用机器学习方法预测某些生产参数或潜在故障，并且他们有一个包含温度和压力数据的数据集。

记录	温度(°C)	压力(bar)
1	50	10
2	55	11
3	53	9.5
…		

- 其中，温度范围为40°C到60°C，压力范围为5 bar到15 bar。

案例：化工生产中的温度与压力数据归一化

- 为了确保这两个参数在机器学习模型中具有相同的权重，工程师决定使用Min-Max缩放将这两个参数的范围归一化到[0, 1]。
- 应用Min-Max缩放后的数据可能是这样的：

记录	温度(归一化)	压力(归一化)
1	0.5	0.5
2	0.75	0.6
3	0.65	0.45
…		

- 这两个参数都在[0, 1]的范围内，可以被机器学习模型公平地对待。

规范化：标准化

- z-score标准化（也称为标准分数标准化或零均值标准化）是一种数据归一化技术，旨在调整特征的值，使其平均值为0，标准差为1。这种方法对于许多统计方法和机器学习算法（如逻辑回归、支持向量机和神经网络）非常有用，因为这些算法通常假设输入特征是零均值和单位方差的。

- z-score的计算公式如下：

$$z = \frac{X - \mu}{\sigma}$$

μ特征的平均值　　σ特征的标准差

- 考虑了特征的均值和方差，因此通常比Min-Max缩放更不易受到异常值的影响。
- 但如果原始数据不是正态分布的，那么标准化后的数据也不会是正态分布的。标准化后的数据不再具有原始数据的物理意义，这可能会使结果解释变得困难。

案例：原料纯度与产量数据的z-score标准化

- **背景**： 在一个化工生产流程中，原料的纯度和生产的产量是两个关键参数。高的原料纯度可能导致高的产品产量。为了分析原料纯度与产量之间的关系，以及为了使用机器学习模型预测未来的产量，工程师收集了一系列的数据。

记录	原料纯度(%)	产量(kg)
1	98	500
2	97	490
3	99	505
…		

- 其中，原料纯度的范围在95%到100%之间，产量的范围在450kg到550kg之间。

案例：原料纯度与产量数据的z-score标准化

- 为了确保原料纯度和产量在分析和机器学习模型中具有相同的权重，工程师选择使用z-score标准化方法对数据进行标准化。
- 例如，假设原料纯度的平均值为97%且其标准差为1%，产量的平均值为500kg且其标准差为20kg。

记录	原料纯度(标准化)	产量(标准化)
1	1.0	0.0
2	0.0	-0.5
3	2.0	0.25
…		

- 原料纯度和产量都被转化为z-scores，这使得它们在后续的分析中具有相似的尺度和分布特性。

离散化

- 将连续的数值特征转换为离散的类别特征。简而言之，离散化就是将连续的数据区间分割成一系列的离散的、不重叠的子区间，并为每个子区间分配一个标签或类别。

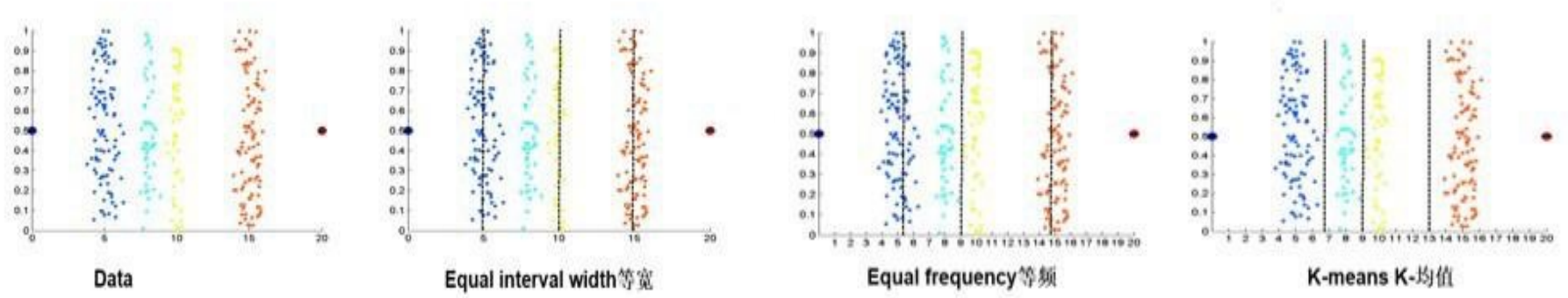
- 为什么要离散化：
  - 简化数据
  - 特定的算法要求：某些机器学习算法（如决策树）在内部进行数据的离散化。手动离散化可以提供更多的控制，以满足特定的业务需求或偏好。
  - 创建有意义的分组：在某些业务场景中，离散化可以帮助创建有实际意义的分组或类别，例如将年龄分为“儿童”、“青少年”、“成年”等。

离散化方法

- **等宽离散化：**将连续属性的值域分割成具有相同宽度的区间的方法。每个区间的宽度是：(最大值 - 最小值) / 区间的数量。所有数据点都被分配到它们所属的区间中。
  - 考虑一组年龄数据：[15, 20, 25, 30, 50, 55, 60]，我们想将这些数据离散化到三个等宽区间中，区间可能是：(10, 30]、(30, 50]、(50, 70]。
- **等频离散化：**将连续属性的值域分割成包含大致相同数量数据点的区间的方法。这种方法尤其在数据的分布不均匀时很有用，因为它确保每个区间中都有相同数量的观察值。考虑一
  - 考虑同样的年龄数据：[15, 20, 25, 30, 50, 55, 60]，我们想将数据离散化到三个等频区间中，可能的区间是：[15, 20, 25]、[30, 50]、[55, 60]，每个区间包含2或3个数据点。

离散化方法

- **基于聚类的离散化：**使用聚类算法（如K-Means）来将连续数据划分为若干个簇，然后每个簇被视为一个离散的区间。此方法的优点是它可以识别并保留数据中的自然结构。
  - 考虑年龄数据：[15, 20, 25, 30, 50, 55, 60]。使用K-Means聚类算法并设置K=3，可能会得到三个簇：[15, 20, 25]、[30]和[50, 55, 60]。每个簇可以看作是一个离散的区间。



特征构造

- **特征提取**：从原始数据中提取新特征的过程，这些新特征能够最有效地表示原始数据中的信息。这通常是为了降低数据的维度、减少计算成本或提高模型的性能。常见的特征值提取方法包括主成分分析（PCA）、线性判别分析（LDA）和自动编码器等。
  （涉及机器学习，后续后进一步阐述）

- **属性构造**：是在数据预处理过程中，从给定的属性集合生成新属性或特征的过程。这种新的属性往往能更好地帮助数据分析或模型的预测。属性构造的主要目标是通过创建新的属性或特征来增加数据的描述能力或提高模型的性能。

特征提取案例

- **背景：**假设一个化工厂正在生产某种化学品。过程中，工程师会监测与生产相关的多个参数，如温度、压力、原料的流速、催化剂的含量、机器的振动等。

  - **原始数据**：每个生产批次，工程师会收集上述各个参数的数据。随着时间的推移，这将产生一个大的数据集，其中每一行代表一个生产批次，每一列代表一个参数。
  - **特征提取的需求**：工程师希望根据这些参数预测生产的质量或产量。但是，由于存在大量的参数，直接使用所有参数可能会导致计算成本过高，或者模型可能会过拟合。
  - **特征值提取**：为了解决这个问题，工程师可以使用PCA或其他方法来提取重要的特征。例如，通过PCA，原始的10个参数可能被压缩到3个主成分，这3个主成分包含了原始数据中的大部分信息。
  - **应用**：使用这3个主成分作为输入，工程师可以构建一个模型来预测产品的质量或产量，而不是使用所有的原始参数。

- 通过特征值提取，工程师得以使用更少的特征（即主成分）来建模，从而降低了计算成本并提高了模型的泛化能力。

属性构造案例

- **背景：**在一个石化工厂中，工程师们使用传感器收集了一个连续反应器的各种运行参数，如温度、压力、流速、原料浓度和反应产物的浓度。

- 基于温度和压力的属性构造：
  - 温度/压力比率：这可能是一个重要的参数，因为它可以表示反应器内部的某种平衡状态，对某些反应可能非常关键。
  - 温度变化率：通过比较当前温度与前一个时间点的温度，可以得到温度变化的速率。这可能会帮助识别反应器是否快速加热或冷却。
- 基于流速和原料浓度的属性构造：
  - 进料量：流速乘以原料浓度可以得到实际进入反应器的原料量。这是一个关键参数，因为它直接影响到反应的速率和效率。
- 基于时间序列数据的属性构造：
  - 滑动平均温度：可以计算过去一定时间段内的平均温度，这有助于了解反应器的热稳定性。
  - 产物浓度的增长速率：这可以通过比较当前的产物浓度与之前的浓度来计算，有助于了解反应速率的变化。
- 组合多个参数的属性构造：
  - 效率因子：可能是基于温度、压力、流速和产物浓度的一个组合值，代表了反应器的整体运行效率。

数据归约

数据归约

- 通过产生一个与原始数据等价或近似的简化版本来代替原始数据。数据归约的目标是减少数据的量，同时确保新的、简化的数据集在后续的数据分析中与原始数据集产生相似或相同的结果。数据归约不仅能减少所需的存储空间，还能加速数据挖掘过程。

- **目的：**
  - **提高效率**：处理简化的数据集可以更快，特别是当处理大数据集时。
  - **减少存储需求**：存储简化的数据集需要更少的存储空间。
  - **更容易进行数据分析**：简化的数据集更容易理解和解释。

数据归约的一些策略

- **维度减少**：如主成分分析（PCA）和线性判别分析（LDA），它们可以减少数据的特征数量。
- **二值化和量化**：将连续数据转化为二进制或有限数量的离散值。
- **直方图、聚类、抽样**：这些技术可以用于减少数据点的数量。
- **数据立方体聚合**：适用于多维数据。
- **回归模型**：用于估算数据和减少数据量。
- **数据压缩技术**：例如使用小波变换进行数据的近似。

数据归约的时间不应当超过或"抵消"
在归约后的数据挖掘/机器学习节省的时间。

维度减少

- 数据归约中的维度减少是指将数据集中的高维特征空间转化为较低维的特征空间，而在此过程中尽量保留原始数据的重要信息。维度减少的目的是减少计算的复杂性、消除冗余或不相关的特征，以及提高数据可视化的效果。

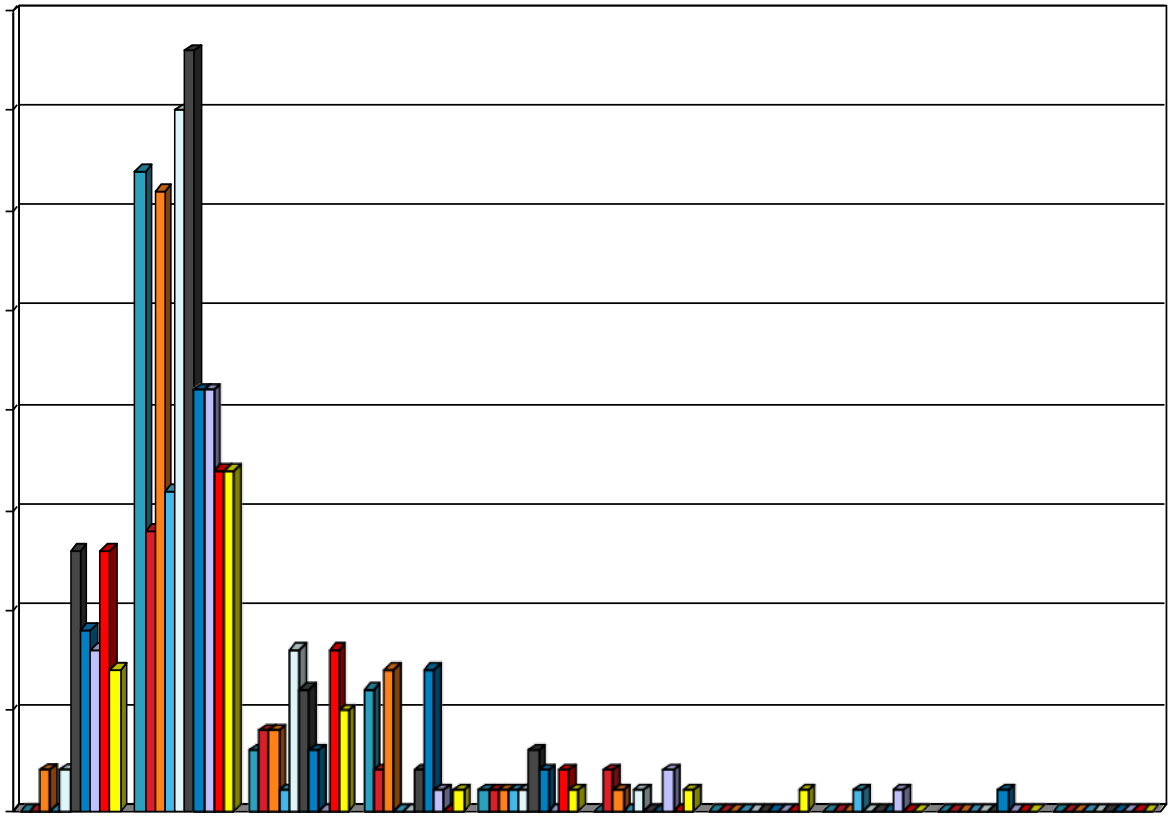
- 例如：假设一个化工公司运营多个生产线，生产各种化学产品。为了监控生产过程并确保质量，每个生产线都安装有众多的传感器，这些传感器每秒都会记录多个参数，例如温度、压力、湿度、流量等。因此，每条生产线每天都会产生大量的数据。在这样的场景下，可能会出现以下问题：
  - **数据冗余**：由于传感器之间的物理相近性或功能相似性，一些传感器可能会产生高度相关或几乎相同的数据。
  - **计算复杂性**：处理和分析这些高维数据需要大量的计算资源和时间。
  - **难以可视化**：对于人类来说，可视化和理解超过三或四个维度的数据是非常困难的。
- 为了解决这些问题，公司可以使用维度减少技术，例如主成分分析（PCA）。PCA 可以找到原始数据中的主要变化方向，并按照这些方向的重要性对其进行排序。通过仅使用少数主成分，可以捕捉到原始数据的大部分信息。

二值化和量化

- **二值化 (Binarization):**
  - 将数据转化为两个值（通常是1和0或者True和False）的过程。它常用于将连续或者多值的属性转化为二值属性。
  - 应用场景：例如，在一个数据集中，我们有一个连续的温度属性。通过设置一个阈值，比如35°C，我们可以将温度二值化为“高”（如果温度大于35°C，赋值为1）和“低”（如果温度小于或等于35°C，赋值为0）。
- **量化 (Quantization):**
  - 定义：量化是将连续的数值属性转化为离散的区间或“桶”的过程。每个区间或“桶”代表了一系列的连续值。
  - 应用场景：再次考虑上述的温度属性。通过量化，我们可以将温度范围划分为几个区间，如0-10°C、10-20°C、20-30°C等，并将每个温度值归入相应的区间。这样，一个连续的温度值就被转化为了一个离散的区间标签。
- **为什么要进行二值化和量化?**
  - 数据简化：通过减少数据的复杂性，使其更容易分析和解释。
  - 存储效率：离散化数据通常需要的存储空间更少。
  - 数据的稳定性：对噪声和异常值更为稳健。
  - 满足特定算法的需求：一些数据挖掘算法更适合处理离散数据。
- 不过，值得注意的是，在进行二值化和量化时，一些细节信息可能会丢失。因此，选择适当的技术和参数以最小化信息损失是非常关键的。

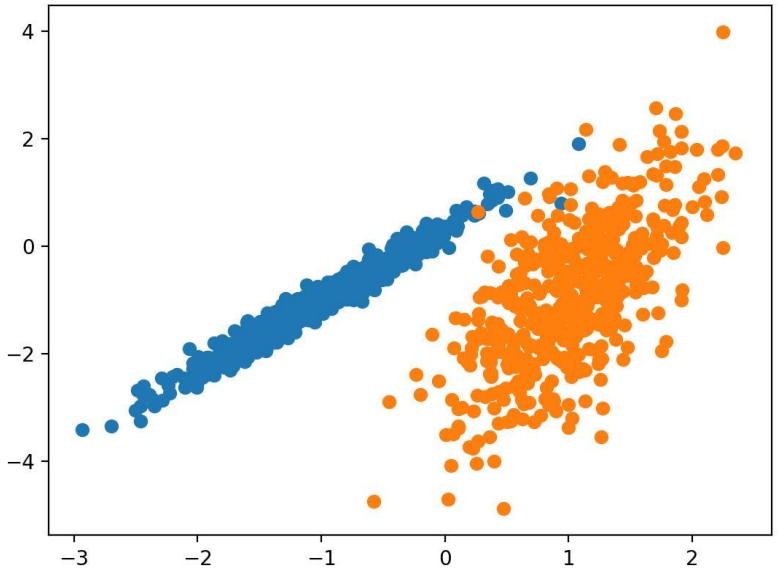
直方图

一种用于表示数据分布的统计工具。它通过将数据分为连续的“桶”或区间并计算每个桶中的数据数量来工作。桶的数量、边界和宽度可以预先定义或基于数据自动计算。直方图的垂直轴表示每个桶中的数据数量，而水平轴表示桶的范围。



聚类

- 将数据集划分为聚类，然后通过聚类来表示数据集
- 如果数据可以组成各种不同的聚类，则该技术非常有效，反之如果数据界线模糊，则方法无效。
- 数据可以分层聚类，并被存储在多层索引树中。



聚类

抽样

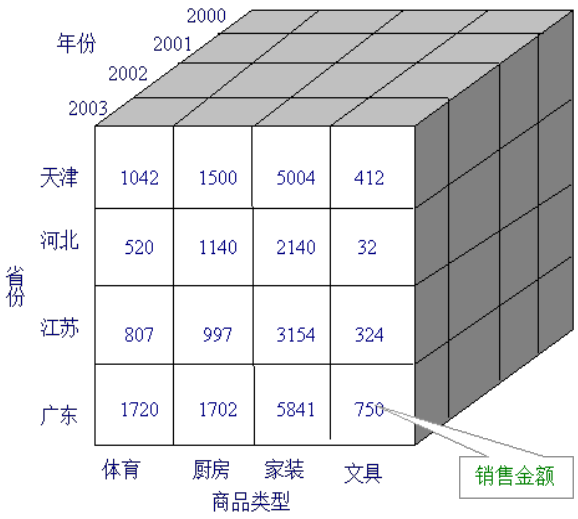
- 从大型数据集中选择一个代表性的子集，旨在减少数据的规模，同时尽量保留数据的结构和分布。抽样的结果可以用于估计、建模或其他数据分析任务。

  - s个样本无放回简单随机抽样：由D的N个元组中抽取s个样本（s<N）
  - s个样本有放回简单随机抽样：过程同上， 只是元组被抽取后，将被回放，可能再次被抽取
  - 聚类抽样：D中元组被分入M个互不相交的聚类中，可在其中的s个聚类上进行简单随机选择
  - 分层抽样：D被划分为互不相交的“层”，则可通过对每一层的简单随机抽样得到D的分层抽样

数据立方体和多维度聚合

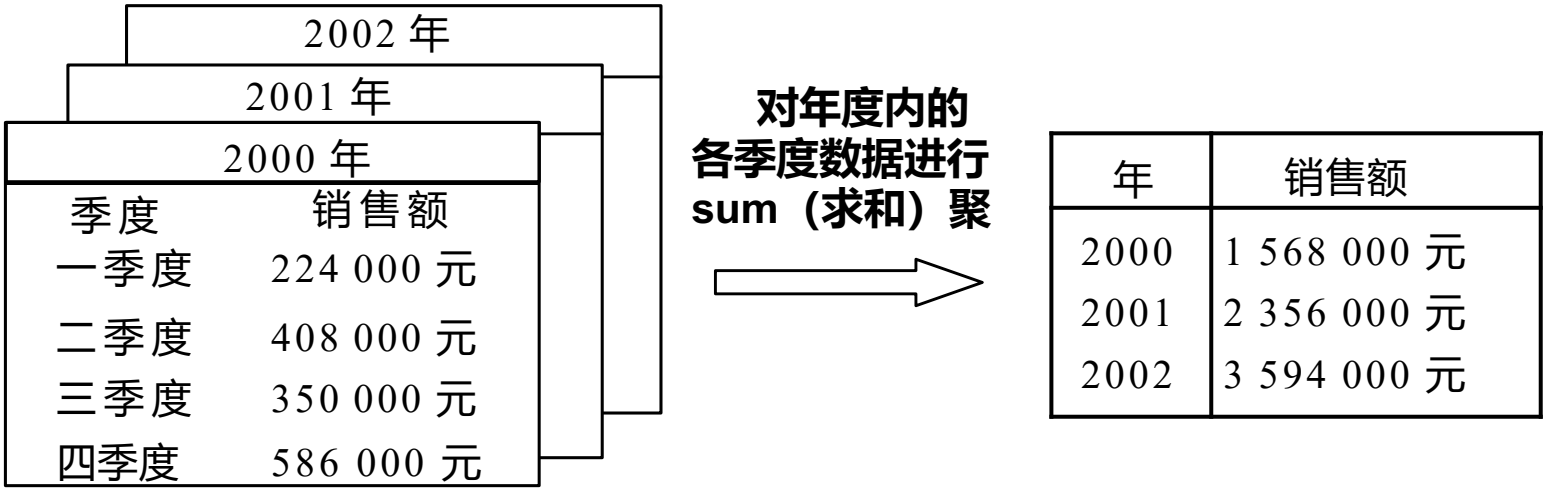
❖ 构建和使用数据立方体或多维数据模型来从多个角度聚集数据，以**便于分析和查询**。

- **数据立方体**：数据立方体是一种多维数据库结构，其中数据沿着每个维度都被组织起来。常见的维度包括时间、地理位置、产品类别等。立方体的每个单元存储了一组数据项的聚合信息，如总和、平均值、计数等。
- **聚集操作**：在数据立方体中，可以对数据进行不同级别的聚集操作。例如，可以聚集按天、按月或按年的销售数据；或者可以聚集某一城市、某一省份或整个国家的销售数据。
- **多维分析**：数据立方体支持切片、切块、钻取、滚动等多维数据分析操作。用户可以从不同的维度和不同的层次查看数据，更深入地分析和理解数据。
- **性能优化**：使用数据立方体聚集，可以预先计算和存储聚合数据，从而提高查询和报告的性能。这是因为查询时直接从预聚合的立方体中获取结果，而不需要实时计算。
- **冗余和存储考虑**：虽然数据立方体提供了快速查询的能力，但它可能会引入一定的数据冗余，并需要更多的存储空间。这是因为立方体可能存储多级聚合数据。



数据聚合

某商场2000～2002年每季度的销售数据，对这种数据进行聚集，使结果数据汇总每年的总销售额，而不是每季度的总销售额。

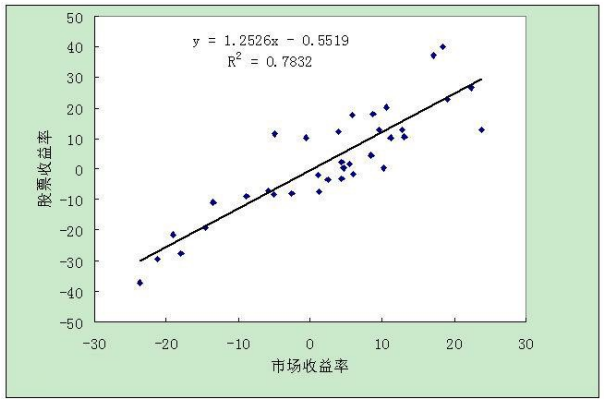


聚集后数据量明显减少，但没有丢失分析任务所需的信息。

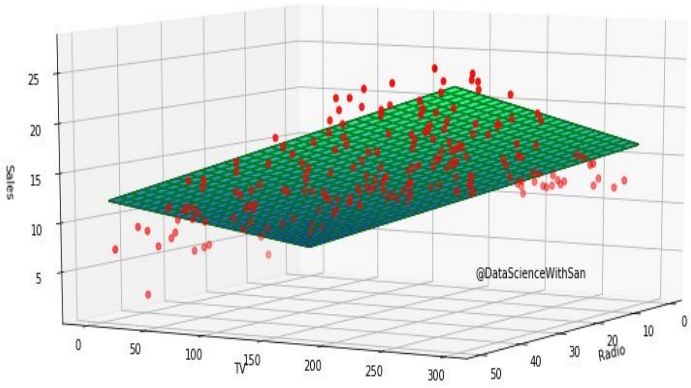
回归分析

- 线性回归：数据被拟合为一条直线$Y = wX + b$
  - 两个回归系数，w和b，由手头的数据来进行估算
  - 通常适用最小二乘法来确定这条直线

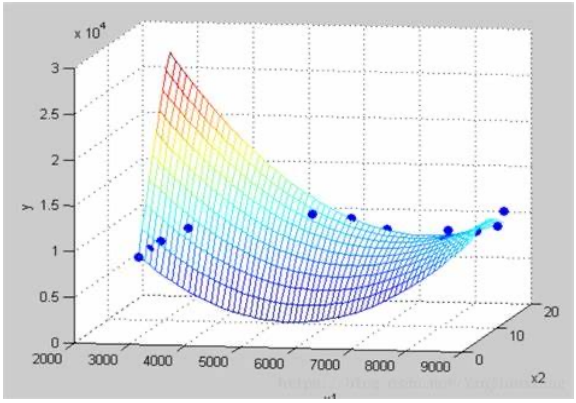
- 多元回归：线性回归的扩充，允许响应变量Y被建模为两个或多个预测　变量的线性函数$Y = b_0 + b_1X_1 + b_2X_2$
  - 多元回归可以拟合多种非线性函数



一元线性回归



多元线性回归



多元非线性回归

数据压缩

- 用**数据编码**或者变换，得到原始数据的压缩表示。
  - 无损(loseless)压缩：可以不丢失任何信息地还原数据。

    如字符串压缩
    - 游程编码： AAAABBBCCDAA ➔ 4A3B2C1D2A
    - 霍夫曼编码：ABRACADABRA"，其中'A'出现5次，'B'和'R'各出现2次，而'C'和'D'各出现1次，那么'A'可能会被分配一个较短的编码，而'C'和'D'可能会得到较长的编码。
    - 字典编码：考虑字符串"ABABABABA"。在LZW压缩中，可以建立一个字典，其中"AB"、"ABA"等都有自己的短编码。然后，将字符串中的重复片段替换为对应的短编码，从而实现压缩。

数据压缩

- 用**数据编码**或者变换，得到原始数据的压缩表示。
  - 有损(lossy)压缩：只能重新构造原数据的近似表示

    图片/音频/视频压缩

    - JPEG（Joint Photographic Experts Group）：这是一种常见的用于图像压缩的方法。它通过丢弃某些视觉上不那么重要的信息（如图像中的高频细节）来实现压缩。通过调整压缩比例，用户可以在文件大小和图像质量之间做出权衡。
    - MP3: 这是一种流行的音频压缩格式。MP3通过丢弃人耳听不到的音频频率和其他技术来实现压缩，从而大大减少文件大小，同时保留了对大多数人来说仍然听起来不错的音质。
    - MPEG（Motion Picture Experts Group）：用于视频压缩的一组标准。与JPEG类似，MPEG通过丢弃某些视觉上不重要的信息来减少文件大小。
    - AAC (Advanced Audio Codec): 比MP3更先进的音频编码格式。它能提供更好的音质和更高的压缩率。

稀疏矩阵

- 稀疏矩阵是其大部分元素为0（或者为某个默认值）的矩阵。在实际应用中，如文本处理、计算机视觉和科学计算中，经常会遇到稀疏矩阵。直接存储这样的矩阵会浪费大量的内存空间，因为大部分的值都是0或默认值。

```
import numpy as np
from scipy.sparse import csr_matrix

# 创建一个2D numpy数组
arr = np.array([
    [0, 0, 3],
    [4, 0, 0],
    [0, 6, 0]
])

# 将numpy数组转换为CSR格式的稀疏矩阵
sparse_mat = csr_matrix(arr)

print(sparse_mat)
```



```
(1, 0)    4
 (2, 1)    6
 (0, 2)    3
```