Data Cleaning Summary

数据清洗 (Data Cleaning) 是指检测并纠正数据中的错误、不一致性和缺失值，以提高数据质量。

常见问题 (Common Issues):

- **不准确数据 (Inaccurate Data)**: 包含噪声、错误或异常值 (Noise, Errors, Outliers)，例如负工资值。
- **不完整数据 (Incomplete Data)**: 缺少属性值或感兴趣的属性 (Missing Attribute Values)。
- **不一致数据 (Inconsistent Data)**: 属性值之间存在差异 (Discrepancies)，例如 "Bill Gates" 和 "B. GATES"。
- **有意错误 (Intentional Errors)**: 使用占位符值 (Placeholder Values)，如 "01/01/1970" 表示未知生日。

处理技术 (Handling Techniques):

1. **不准确数据 (Inaccurate Data)**:
    - 平滑方法 (Smoothing Methods): 分箱 (Binning)、回归 (Regression)、聚类 (Clustering)。
2. **不完整数据 (Incomplete Data)**:
    - 忽略元组 (Ignore Tuples)、手动填充 (Fill Manually)、或使用自动方法 (Automated Methods)，如均值 (Mean)、众数 (Mode)、或基于推断 (Inference-Based)。
3. **不一致数据 (Inconsistent Data)**:
    - 使用元数据分析 (Metadata Analysis)、规则检查 (Rule Checking)、领域知识 (Domain Knowledge) 检测并解决问题。

过程 (Process):

- **检测 (Detection)**: 分析元数据、验证数据库结构、检查规则。
- **纠正 (Correction)**: 应用适当的技术清洗数据。

目标 (Goal):

确保数据的准确性 (Accuracy)、完整性 (Completeness)、一致性 (Consistency)，以支持可靠的分析。

Data Reduction Summary

数据降维 (Data Reduction) 是通过减少数据集的表示形式来降低数据的复杂性，同时尽量保持分析结果的准确性。处理大规模数据集，减少复杂性。降低存储成本，加快分析速度。聚焦于最相关的信息。

1. **维度约简 (Dimensionality Reduction)**: 减少考虑的变量数量，避免维度灾难 (Curse of Dimensionality)。
    - (Principal Component Analysis, PCA)
    - 属性子集选择 (Attribute Subset Selection)
    - 属性构造 (Attribute Construction)。
2. **数量约简 (Numerosity Reduction)**: 使用替代的、更小的数据表示形式。
    - 参数方法 (Parametric Methods): 假设数据符合数学模型，仅存储模型参数。
    - 非参数方法 (Non-parametric Methods): 使用直方图 (Histograms)、聚类 (Clustering)、采样 (Sampling) 等。
3. **数据压缩 (Data Compression)**:

- 无损压缩 (Lossless Compression): 保留所有信息。
- 有损压缩 (Lossy Compression): 允许部分信息丢失以换取更高的压缩率。
- **目标 (Goal)**: 在尽量减少信息损失的情况下，获得更小的数据表示形式以支持高效的数据挖掘。

Attribute Subset Selection

属性子集选择 (Attribute Subset Selection) 是一种特征选择方法，通过选择原始数据中最相关的属性子集，去除冗余或无关的属性，从而提高数据挖掘的效率和准确性。减少数据维度，避免维度灾难 (Curse of Dimensionality)。提高模型的性能和可解释性。降低计算成本和存储需求。消除噪声和冗余信息。

1. **冗余属性 (Redundant Attributes)**:
   - 提供重复信息的属性，例如商品价格和税额。
2. **无关属性 (Irrelevant Attributes)**:
   - 对目标任务没有影响的属性，例如学生 ID 对 GPA 的预测。

方法 (Methods):

1. **启发式搜索 (Heuristic Search)**:
   - **前向选择 (Forward Selection)**: 从空属性集开始，每次迭代选择最优的剩余属性加入。
   - **后向消除 (Backward Elimination)**: 从完整属性集开始，每次迭代移除最差的属性。
   - **决策树归纳 (Decision Tree Induction)**: 使用决策树的非叶节点作为测试属性，选择最优分裂点。
2. **评估标准 (Evaluation Metrics)**:
   - 使用信息增益 (Information Gain)、卡方检验 (Chi-Square Test)、或相关性分析 (Correlation Analysis) 来评估属性的重要性。
3. **注意事项 (Notes)**: 属性子集选择是一个 NP 难问题，穷举搜索所有可能的属性组合通常不可行; 启发式方法提供了高效的近似解; 选择的属性子集应与目标任务高度相关，同时避免过拟合。

示例 (Example):

- 假设一个数据集包含以下属性：学生 ID、年龄、性别、GPA、电话号。
- 目标是预测 GPA。
- 属性子集选择可能会去除学生 ID 和电话号，因为它们与 GPA 无关。

Principal Component Analysis, PCA

主成分分析 (PCA) 的目标是通过线性变换将高维数据投影到低维空间，同时尽可能保留数据的方差信息。

- 降低数据维度，同时尽量保留数据的主要信息。
- 消除变量之间的相关性。
- 提高数据挖掘和机器学习算法的效率。

核心思想 (Core Idea):

- 找到一组正交基向量（主成分），使得数据在这些基向量上的投影具有最大的方差。
- 主成分按其解释的方差大小排序，前几个主成分保留了数据的主要信息。

数学步骤 (Mathematical Steps):

1. **数据标准化 (Data Standardization)**: 将每个特征的均值调整为 0，标准差调整为 1。标准化公式：$x' = \frac{x-\mu}{\sigma}$ 其中 μ 是均值，σ 是标准差。
2. **计算协方差矩阵 (Covariance Matrix)**: 协方差矩阵衡量特征之间的线性关系：$\mathrm{Cov}(X) = \frac{1}{n-1}X^T X$ 其中 X 是标准化后的数据矩阵。
3. **特征值分解 (Eigen Decomposition)**:
   - 对协方差矩阵进行特征值分解，得到特征值 λ 和特征向量 v：
   - $\mathrm{Cov}(X)v = \lambda v$
   - 特征值 λ 表示主成分的方差大小。
   - 特征向量 v 表示主成分的方向。
4. **选择主成分 (Select Principal Components)**:按特征值从大到小排序，选择前 k 个特征向量作为主成分。
5. **数据投影 (Project Data)**:将原始数据投影到选定的主成分上：$Z = XV_k$ 其中 V_k 是前 k 个特征向量组成的矩阵，Z 是降维后的数据。

几何解释 (Geometric Interpretation):

- 主成分是数据分布方差最大的方向。
- 每个主成分与其他主成分正交（线性无关）。

注意事项 (Notes):

- PCA 假设数据的主要信息体现在方差中，因此对噪声敏感。
- 数据标准化是必要的，尤其当特征的量纲不同。
- PCA 仅适用于数值型数据。
- 主成分的选择需要权衡信息保留和降维效果。
- 数据标准化是关键步骤，避免变量范围差异影响结果。

公式总结 (Summary of Key Formulas):

- 协方差矩阵：$\mathrm{Cov}(X) = \frac{1}{n-1}X^T X$
- 特征值分解：$\mathrm{Cov}(X)v = \lambda v$
- 数据投影：$Z = XV_k$

实现 (Implementation):

```
import numpy as np
from sklearn.decomposition import PCA

# 示例数据
data = np.array([[2.5, 2.4], [0.5, 0.7], [1.1, 0.9]])
# skip normalization
# 使用 sklearn 实现 PCA
pca = PCA(n_components=2)  # 指定主成分数量
reduced_data = pca.fit_transform(normalized_data)

print("降维后的数据:",  reduced_data)
print("主成分方向 (特征向量):",  pca.components_)
print("主成分的重要性 (特征值):",  pca.explained_variance_ratio_)
```

Numerosity Reduction

数量约简 (Numerosity Reduction) 是一种数据降维技术，通过替代的、更小的数据表示形式来减少数据体积，同时尽量保留数据的主要信息。

目的 (Purpose):

- 减少数据存储需求。
- 提高数据挖掘和分析的效率。
- 降低计算复杂度。

方法分类 (Methods):

1. **参数方法 (Parametric Methods)**:
   - 假设数据符合某种数学模型，仅存储模型参数，丢弃原始数据（可能保留异常值）。
   - 示例：
   - **回归分析 (Regression Analysis)**:
     - 使用线性或非线性回归模型拟合数据。
     - 例如，线性回归公式：$Y = wX + b$，其中 w 和 b 是回归系数。
   - **时间序列模型**: 使用 ARIMA 等模型预测数据趋势。
2. **非参数方法 (Non-parametric Methods)**:
   - 不假设数据的分布或模型，直接使用替代表示形式。
   - **直方图 (Histograms)**:
     - 将数据划分为不相交的桶（bins），每个桶存储平均值或总和。
   - **聚类 (Clustering)**:
     - 将数据分组为相似的簇，仅存储簇的中心点（质心）或其他表示。
   - **采样 (Sampling)**:
     - 从数据集中抽取一个小样本来代表整个数据集。

注意事项 (Notes):

- 参数方法适用于数据符合已知模型的情况，但可能丢失非模型化的信息。
- 非参数方法更灵活，但可能需要更多的存储空间。
- 选择方法时需权衡数据表示的简洁性与信息保留的完整性。

示例 (Example):

- **直方图**:
  - 数据：$1, 1, 5, 5, 5, 8, 10, 10, 12$
  - 等宽分箱 (Equal-Width Binning)：分为 3 个桶，范围分别为 $1-5, 6-10, 11-15$。
  - 每个桶的平均值：$3, 7.5, 12$。
- **聚类**:
  - 数据点：$(1, 2), (2, 3), (10, 10), (11, 11)$
  - 聚类结果：两簇，质心分别为 $(1.5, 2.5), (10.5, 10.5)$。

Data Transformation Summary

数据转换 (Data Transformation) 是将原始数据映射为新的替代值集合，以便更适合数据挖掘任务。

目的 (Purpose):

- 统一数据格式，消除不一致性。
- 提高数据挖掘算法的效率和准确性。
- 支持特定的挖掘任务（如相似性计算、分类等）。

策略 (Strategies):

1. **归一化 (Normalization)**: 将数据缩放到较小的范围（如 [0, 1] 或 [-1, 1]）。
    - **Min-Max 归一化**
    - **Z-score 标准化**
    - **小数缩放 (Decimal Scaling)**
2. **平滑 (Smoothing)**:
    - 去除数据中的噪声。
    - 方法：分箱平滑、回归平滑、聚类平滑。
3. **聚合 (Aggregation)**:汇总数据，例如在数据立方体中进行维度聚合。
4. **属性构造 (Attribute Construction)**: 基于现有属性创建新属性，例如通过“数量 × 单价”计算“总成本”。

注意事项 (Notes):

- 数据转换应根据具体任务需求选择合适的方法。
- 归一化对基于距离的算法（如 KNN、K-means）尤为重要。
- 离散化可以减少数据复杂性，但可能丢失部分信息。

Normalization Techniques

1. Min-Max Normalization

1. 确定属性的最小值 (minA) 和最大值 (maxA)。
2. 选择新的范围 $(\text{new\_minA}, \text{new\_maxA})$。
3. 使用公式将原始值映射到新范围。
4. **公式 (Formula)**: $\text{v}' = \frac{\text{v}-\text{minA}}{\text{maxA}-\text{minA}} \times (\text{new\_maxA} - \text{new\_minA}) + \text{new\_minA}$
5. **联系 (Relationship)**: 依赖于数据集的最小值和最大值；不考虑数据的分布；对异常值（outliers）敏感，因为极值决定了缩放范围。
6. **使用场景 (Use Cases)**: 适用于需要固定范围的数据（如图像处理、神经网络）；适合对绝对值敏感的算法（如 KNN、K-means）。

实例 (Example):

- 原始数据：$12,000, 73,600, 98,000$
- 范围：$12,000, 98,000$ 映射到 $0, 1$。
- 对 $\text{v} = 73,600$： $\text{v}' = \frac{73,600-12,000}{98,000-12,000} \times (1 - 0) + 0 = 0.716$

2. Z-Score Normalization

1. 计算属性的均值 (μ) 和标准差 (σ)。
2. 使用公式将原始值标准化。
3. **公式 (Formula)**: $v' = \frac{v-\mu}{\sigma}$
4. **联系 (Relationship)**: 相较于 Min-Max，对异常值不太敏感；依赖于数据的均值 (μ) 和标准差 (σ)；将数据标准化为均值为 0，标准差为 1。考虑数据的分布。
5. **使用场景 (Use Cases)**: 数据服从高斯分布时效果较好； 适用于假设数据正态分布的算法（如 PCA、线性回归）。

实例 (Example):

- 原始数据：$54,000, 73,600, 98,000$
- 均值：$\mu = 54,000$，标准差：$\sigma = 16,000$。
- 对 $v = 73,600$： $v' = \frac{73,600-54,000}{16,000} = 1.225$

3. Decimal Scaling

1. 找到数据的最大绝对值 ($\max(|v|)$)。
2. 确定缩放因子 j，使得 $\max(|v|)/10^j < 1$。
3. 使用公式将原始值缩放。
4. **公式 (Formula)**:$v' = \frac{v}{10^j}$
5. **联系 (Relationship)**: 通过移动小数点将数据缩放，基于最大绝对值; 简单且计算效率高; 不考虑数据的分布。
6. **使用场景 (Use Cases)**: 适用于已知数据范围且需要简单缩放的场景; 适合对解释性要求较高的任务。

实例 (Example):

- 原始数据范围：$-986, 917$。
- 最大绝对值：$\max(|v|) = 986$，因此 $j = 3$。
- 对 $v = 917$： $v' = \frac{917}{10^3} = 0.917$
- 新范围：$-0.986, 0.917$。

Data Discretization

数据离散化 (Data Discretization) 是将连续属性的值划分为有限的离散区间，并用区间标签替换实际数据值，从而减少数据复杂性并提高数据挖掘效率。

- 简化数据表示，减少数据规模。
- 提高数据挖掘算法的效率和准确性。
- 支持分类、聚类等任务。

方法分类 (Methods):

1. **等宽分箱 (Equal-Width Binning)**: 将数据范围划分为等宽的区间。
    - 区间宽度公式： $W = \frac{\max-\min}{N}$

- 其中 N 是区间数量。
- **优点**: 简单易用。**缺点**: 对偏态数据效果较差，可能导致区间不平衡。

2. **等频分箱 (Equal-Frequency Binning)**:
   - 将数据划分为包含相同数量数据点的区间。
   - **优点**: 适合偏态数据。**缺点**: 区间宽度可能不均匀。
3. **基于聚类 (Clustering)**: 使用聚类算法（如 K-means）将数据划分为多个簇，每个簇对应一个区间。
   - **优点**: 能更好地捕捉数据分布。**缺点**: 计算复杂度较高。
4. **基于分类 (Classification)**:
   - 使用监督学习方法（如决策树）根据目标变量划分区间。
   - **优点**: 考虑了目标变量的分布。
   - **缺点**: 需要标注数据。
5. **基于相关性分析 (Correlation Analysis)**: 使用统计方法（如 ChiMerge）合并具有相似分布的相邻区间。
   - **优点**: 考虑了区间之间的相关性。**缺点**: 需要计算统计量。

注意事项 (Notes):

- 离散化方法的选择应根据数据分布和任务需求。
- 等宽分箱适合均匀分布的数据，而等频分箱适合偏态数据。
- 聚类和分类方法适合需要更复杂划分的场景。

示例 (Example):

- 原始数据：$4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34$
- **等宽分箱**（宽度 = 10）：
  - Bin 1: $4, 14) \rightarrow$4, 8, 9$
  - Bin 2: $14, 24) \rightarrow$15, 21, 21$
  - Bin 3: $24, 34 \rightarrow 24, 25, 26, 28, 29, 34$
- **等频分箱**：
  - Bin 1: $4, 15 \rightarrow 4, 8, 9, 15$
  - Bin 2: $21, 25 \rightarrow 21, 21, 24, 25$
  - Bin 3: $26, 34 \rightarrow 26, 28, 29, 34$