模式识别与机器学习

Lecture Notes

笔记四

特征工程与数据准备

目录

1 机器学习数据准备

1.1 数据准备通常包含哪些工作

数据准备通常可以分为四个紧密衔接的部分：数据收集、数据清洗、特征工程和数据集划分。它们在实际项目中往往相互影响、反复迭代，共同决定模型能否被有效训练。

1.1.1 数据收集：决定模型看到怎样的世界

数据收集回答的是”训练数据从哪里来”这一基础问题。模型并不直接理解现实世界，而是通过训练数据去”认识”世界。

在实际任务中，数据来源可能包括业务数据库、日志系统、传感器、人工标注平台、公开数据集或第三方接口。无论来源如何，都应关注：

- 相关性：数据是否与任务目标相关
- 代表性：是否能反映真实应用场景
- 真实性：是否存在系统性偏差
- 合规性：是否符合隐私与法律要求

1.1.2 数据清洗：把原始记录变成可信的数据基础

数据清洗的目标是让数据在统计意义和业务意义上都更可信。现实中的原始数据通常伴随空值、异常值、重复项、格式混乱和错误标注等问题。

数据清洗不是单纯”删除脏数据”，而是：

- 修复信息质量
- 降低噪声干扰
- 统一表达标准
- 为后续特征工程提供可靠基础

1.1.3 特征工程：把数据转化为更能表达问题本质的形式

特征工程的核心是：**把原始数据转化成更适合学习任务的特征表达**。

它通常包括：

- 文本数值化、类别变量编码
- 时间戳拆解（年、月、日、小时等）
- 不同量纲的数值尺度统一
- 从已有字段构造新特征
- 特征筛选与评估

在传统机器学习中，特征工程的质量往往比模型选择更能决定最终效果。

1.1.4 数据集划分：让训练、调参与评估彼此分离

数据集划分的核心目的是让训练、调参和评估相互独立：

- **训练集**：用于学习模型参数
- **验证集**：用于模型选择与超参数调优
- **测试集**：用于最终评估泛化性能

如果划分不当或存在数据泄漏，评估指标可能虚高，无法反映真实应用效果。

1.2 什么是数据清洗

数据清洗是指在建模前，对原始数据中的缺失、错误、冲突、异常和不一致之处进行识别与处理，使其质量满足后续分析与训练要求。

它关心的核心问题是：

- 数据是否真实
- 是否完整
- 是否一致
- 是否可用于建模

1.3 数据清洗的一般流程

数据清洗可以整理为以下七个步骤，它们通常不是严格线性的，而是反复迭代。

1.3.1 数据理解

明确每个字段的含义、数据类型、样本规模和业务约束。只有在理解数据的前提下，后续处理才是可靠的。

1.3.2 质量检查

对数据进行系统扫描，识别缺失、不一致、重复、越界或逻辑冲突问题，并建立问题清单。

1.3.3 缺失处理

判断缺失是随机的、系统性的，还是本身携带业务信息，再选择删除、填补或引入缺失标志变量。

1.3.4 异常处理

区分”错误异常”和”真实极端值”，避免不加区分地删除有价值的稀有样本。

1.3.5 去重

在明确主键和业务对象的基础上，识别并合并真正的重复记录，而不是盲目删除相似样本。

1.3.6 标准化

统一数值尺度、日期格式、单位和文本表达方式，降低数据表达的混乱程度。

1.3.7 输出数据

形成结构清晰、规则统一、处理过程可追溯的数据版本，作为建模阶段的正式输入。

1.4 一个简化案例：从学生成绩表到可训练数据

假设目标是根据作业、考勤、测验和期中成绩预测期末成绩：

1. 收集来自不同系统的原始成绩数据
2. 清洗缺考、超满分、重复导入等问题
3. 构造”作业平均分””出勤率””测验波动程度”等特征
4. 划分训练集、验证集和测试集

这一案例体现了数据准备从原始记录走向可训练样本的完整路径。

1.5 对数据准备的整体理解

数据准备本质上承担了”把现实世界翻译成模型语言”的作用：

- 数据收集：决定模型能看到怎样的现实
- 数据清洗：决定输入是否可信
- 特征工程：决定问题是否被恰当表达
- 数据集划分：决定评估是否科学、公平

2 特征工程基础

2.1 特征质量评估

根据数据分布与模型特性的契合度，一个高质量的特征应当满足以下三个核心条件：**有信息、少噪声、符合模型假设**。

1. 有信息 (Informative) 好的特征必须与目标变量 Y 存在高度的相关性。在实际工程中，我们通常结合业务直觉 (Business Intuition) 与探索性数据分析 (EDA, Exploratory Data Analysis) 来挖掘特征。

在数学上，我们可以使用皮尔逊相关系数 (Pearson Correlation) 或互信息 (Mutual Information) 来量化特征 X 与目标 Y 之间的信息量。互信息的定义如下：

$$I(X;Y) = \sum_{y\in\mathcal{Y}}\sum_{x\in\mathcal{X}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{1}$$

2. 少噪声 (Low Noise) 现实世界的数据通常是" 脏" 的，包含了大量的异常值和量纲差异。数值的绝对大小本身可能会对距离度量（如 KNN 或 SVM）产生误导。因此，我们需要对特征进行缩放以消除量纲带来的噪声干扰。最常用的方法是 Z-Score 标准化 (Standardization)：

$x_{scaled} = \frac{x-\mu}{\sigma}$。

其中 μ 为均值，σ 为标准差。这能使数据分布符合标准正态分布，加速基于梯度下降的算法收敛。

3. 符合模型假设 (Fits Model Assumptions) 好特征不仅要" 有信息"，还必须" 适合模型"。不同的算法对数据分布有特定的假设前提：

- 线性模型 (如线性回归、逻辑回归)：假设特征与对数几率之间存在线性关系。如果现实数据是非线性的（如指数增长），直接输入线性模型效果会很差。

- 树模型 (如随机森林、XGBoost)：对单调非线性变换免疫，但对特征的组合和共线性有不同要求。

针对不符合假设的数据，必须进行特征变换 (Feature Transformation)。例如，对于存在长尾分布（右偏）的特征，我们常采用对数变换来稳定方差，使其更接近正态分布：$x_{transformed} = \log(x + 1)$。

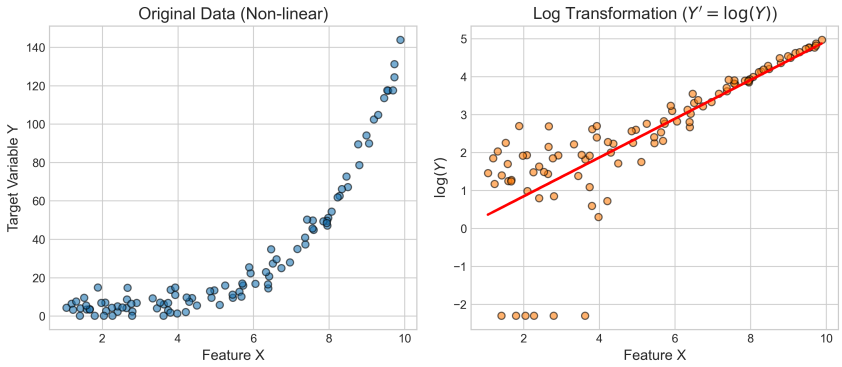

图 1: 特征变换示意图（对数变换）

2.2 特征数量权衡

特征数量的多少直接决定了特征空间的维度，进而深刻影响模型的泛化能力。特征选择本质上是在信息量与模型复杂度之间寻找最优的平衡点。

1. **特征过少：欠拟合风险**

当提取的特征过少时，模型面临信息不足 (Under-representation) 的窘境。

- 表达能力受限：少量的特征无法刻画复杂数据的全貌，导致模型具有高偏差 (High Bias)。
- 任务失败：在面对复杂的非线性分类或回归任务时，极少的特征会导致模型在训练集和测试集上的表现均不理想，即发生欠拟合 (Underfftting)。

2. **特征过多：过拟合与维度灾难**

盲目增加特征数量并非总是益事，这会引入经典机器学习难题——维度灾难 (Curse of Dimensionality)：

- 训练成本剧增：随着特征维度的增加，所需的训练样本量呈指数级增长，同时矩阵运算的计算与内存开销也会急剧增加。

- 模型复杂度提高与过拟合：过多的特征赋予了模型过剩的拟合能力，导致模型具有高方差 (High Variance)。

- 引入噪声干扰：大量冗余或不相关的特征会掩盖真实信号。在极高维度下，样本之间的距离度量（如欧氏距离）会失去区分度，严重干扰模型的学习方向，从而极大降低其在未见数据上的泛化能力 (Generalization Ability)。

综上所述，优秀的特征工程并不是盲目堆砌数据，而是通过降维（如 PCA）或特征选择（如 L1 正则化、递归特征消除）手段，剔除冗余和噪声，保留最核心的表征。

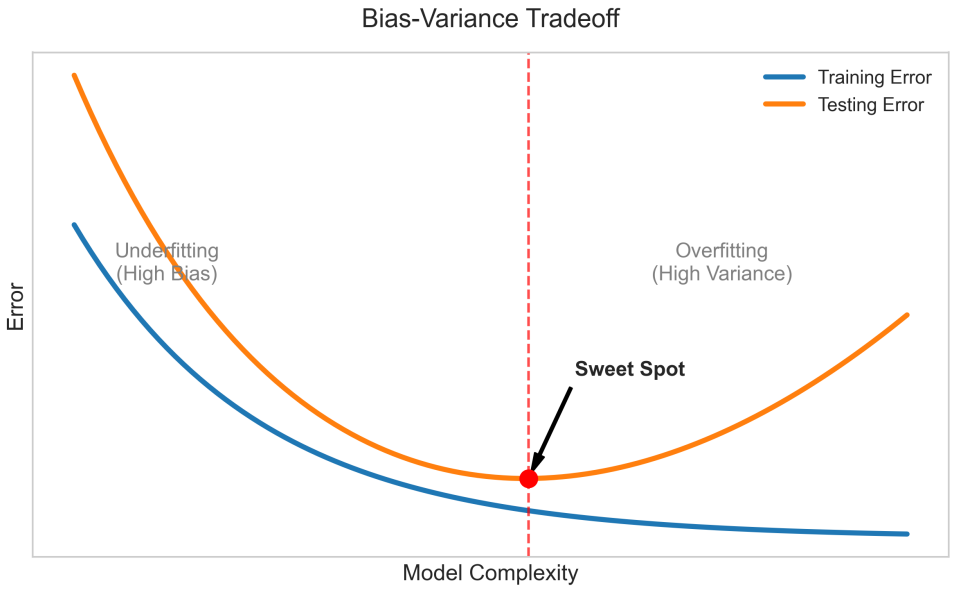

图 2: 特征数量与泛化误差的关系：偏差-方差权衡

2.3 特征工程对模型的作用

1. **提升数据信噪比**

   剔除冗余变量，降低噪声对模型训练的干扰。

2. **赋予模型非线性建模能力**

   通过特征交叉（如 $x_1 \times x_2$）帮助线性模型捕获非线性规律。

3. **优化数值稳定性**

   通过标准化消除量纲差异，改善损失函数的几何形状。

4. **语义对齐**

将业务逻辑（时间、地点、行为序列）转化为模型可计算的数值向量。

2.4 特征工程的标准执行流程

特征工程是一个闭环迭代系统，其标准工作流如下：

$$\text{数据理解} \rightarrow \text{特征构造} \rightarrow \text{特征变换} \rightarrow \text{特征选择} \rightarrow \text{特征评估}$$

2.4.1 数据理解

通过探索性数据分析量化数据的统计特性：

- 均值、方差、偏度、峰度
- 缺失值分布与离群点识别
- 特征共线性分析

2.4.2 特征构造

通过数学变换或逻辑组合产生高阶特征：

- 特征交叉与多项式扩展
- 业务逻辑驱动的属性分解

2.4.3 特征变换

调整特征分布以满足算法假设：

- 无量纲化：标准化（z-score）、极差归一化
- 非线性变换：Log、Box-Cox
- 离散化与 One-Hot 编码

2.4.4 特征选择

筛选对目标变量具有强解释力的特征子集：

- 过滤法（Filter）
- 包装法（Wrapper）
- 嵌入法（Embedded，如 Lasso）

2.4.5 特征评估

定量评估特征对模型性能的贡献：

- 特征重要性评分
- 置换检验（Permutation Test）
- 交叉验证性能对比

2.5 特征构建案例：游戏商城皮肤购买预测

特征构建的本质是：**将隐含在原始数据中的业务逻辑显性化**。

2.5.1 操作逻辑

- **属性组合与混合**

  将多个相关属性进行数学运算，构造反映关联关系的新特征。

- **分解与切分**

  将复杂属性拆解为更细粒度的维度，使模型能捕捉局部规律。

2.5.2 案例分析：游戏商城皮肤购买预测

原始数据：用户背包物品编码序列（如 `12301, 12302, 12403`）**编码规则**：前三位为英雄 ID，后两位为皮肤编号

构建的特征

1. **英雄拥有状态（二值特征）**

   判断背包中是否存在目标英雄 ID（如 123）。

2. **英雄皮肤深度（计数特征）**

   统计用户已拥有该英雄的皮肤数量。

3. **系列皮肤偏好（比率特征）**

   统计特定编号皮肤的占比，反映用户偏好。

4. **重复购买冲突检测（冲突特征）**

   判断用户是否已拥有目标皮肤，避免无效推荐。

3 特征构建与编码

3.1 特征构建概述

特征构建（Feature Construction）是机器学习工作流中的核心环节，是从原始数据中推导出新特征的过程，旨在增强数据的表达能力，使模型能够更有效地捕捉底层规律。

其本质是将非结构化或低表达性的原始记录，转换为模型可理解的数值向量：

- **原始数据**：用户行为日志、文本、图像像素等
- **特征表达**：通过聚合、转换、组合生成的量化指标
- **目标**：提高特征与目标变量之间的相关性，降低模型的学习难度

3.2 实例：Echo Nest 音乐推荐系统

基于 Echo Nest Taste Profile 数据集来演示特征构建的具体过程。

3.2.1 数据背景与原始特征

该数据集记录了大规模用户的音乐听歌历史，包含以下关键维度：

- **用户基数**：约 100 万 (1,019,318)
- **歌曲基数**：约 38 万 (384,546)
- **交互记录**：约 4800 万条
- **原始数据格式**：`(user_id, song_id, listen_count)`,例如 `(user_001, song_123, 15)`

3.2.2 特征构建逻辑：从行为到偏好

在推荐系统中，我们需要将“听歌次数”转换为“用户对歌曲的偏好特征”。

1. **交互矩阵构建：**将原始三元组展开为 $M \times N$ 的矩阵 R，其中 $R_{i,j}$ 表示用户 i 对歌曲 j 的交互权重。

2. **隐式反馈量化：**原始的 `listen_count` 往往存在长尾分布，通常需要进行对数缩放（Log-scaling）或二值化处理，以消除极端值的影响：

$$f(x) = \log(1 + x)$$

3.2.3 相似度挖掘与数学表达

特征构建的最终目标是计算物品或用户之间的相似性。根据“喜欢相似物品的人未来也会喜欢相似物品”的逻辑，我们需要量化相似度。

假设我们将用户 A 和用户 B 对歌曲的喜好构建为特征向量 $\mathbf{u}_A$ 和 $\mathbf{u}_B$，常用的相似度衡量标准是余弦相似度（Cosine Similarity）：

$$\text{Similarity}(\mathbf{u}_A, \mathbf{u}_B) = \frac{\mathbf{u}_A \cdot \mathbf{u}_B}{\|\mathbf{u}_A\|\|\mathbf{u}_B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

若计算得出用户 A 与用户 B 的特征向量高度相似，且：

- 用户 A 喜欢：歌曲 1, 2
- 用户 B 喜欢：歌曲 1, 2, 3

则通过特征匹配，系统会将歌曲 3 推荐给用户 A。

3.3 One-Hot 编码

3.3.1 基本概念

One-Hot 编码（独热编码）是机器学习中处理类别型特征的基础数值化方法，核心是将离散类别映射为正交的二进制向量：对于一个有 N 个可能取值的离散特征，用长度为 N 的二进制向量表示，其中仅一个位置为 1，其余为 0。

例如，“年级”特征的三个取值可编码为：初一 = 100、初二 = 010、初三 = 001；“学校”特征的四个取值可编码为：一中 = 1000、二中 = 0100、三中 = 0010、四中 = 0001。实际应用中，多个类别特征的 One-Hot 向量可按顺序拼接，形成统一的模型输入向量，如性别、年级、学校拼接后会形成长度为 $2+3+4=9$ 的特征向量。

图 3: One-Hot 编码示例表格

3.3.2 为什么不用直接编号

机器学习中，绝大多数算法无法直接处理文本型类别特征，因此需要先将其数值化。但直接对类别编号会引入严重的伪顺序关系问题：模型会将数字的大小与类别关联，错误认为“四中比一中更大”“三中和四中更接近”，而实际上这些学校只是独立类别，不存在这样的数学意义。

One-Hot 编码通过正交映射消除了这种伪顺序关系，让模型只关注类别本身的差异，避免有序编号带来的训练偏差。

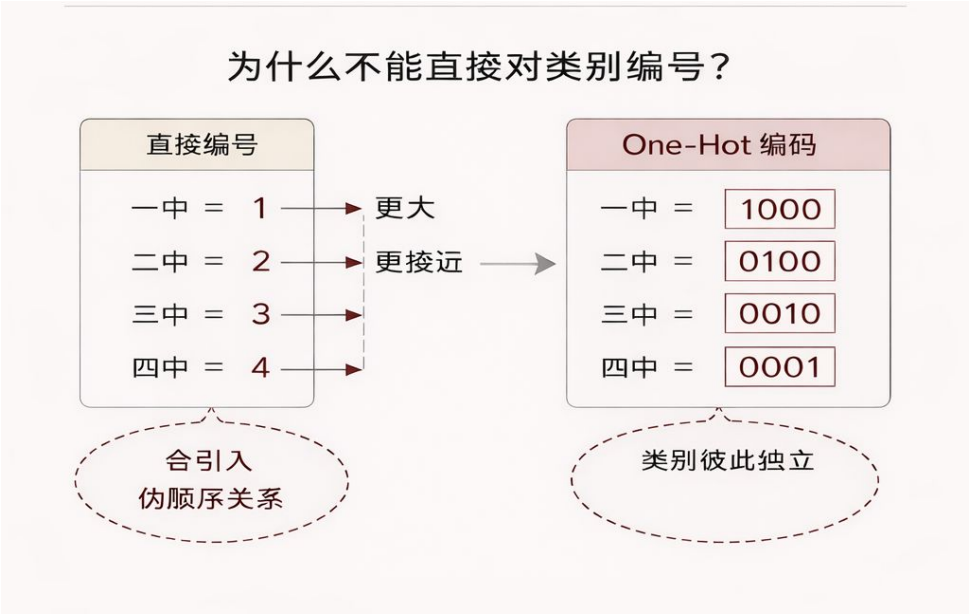

图 4: 直接编号与 One-Hot 编码对比

3.3.3 优点

One-Hot 编码的优势主要体现在以下几点：

1. **不会引入伪顺序关系**：每个类别被映射到独立的正交维度，不存在大小或远近的偏差，完全还原了类别特征的真实分布。

2. **通用性强**：编码后的数值向量格式统一，可直接输入线性模型、神经网络、SVM 等各类算法，无需额外转换。

3. **可解释性强**：每个维度唯一对应一个类别，可通过特征权重直观判断类别对预测结果的影响。

4. **便于多特征拼接**：不同类别的 One-Hot 向量格式统一，可直接拼接成更长的特征向量。

5. **天然适配按位掩码操作**：可通过掩码向量灵活屏蔽或保留特定特征段，在特征消融、注意力机制中尤为实用。

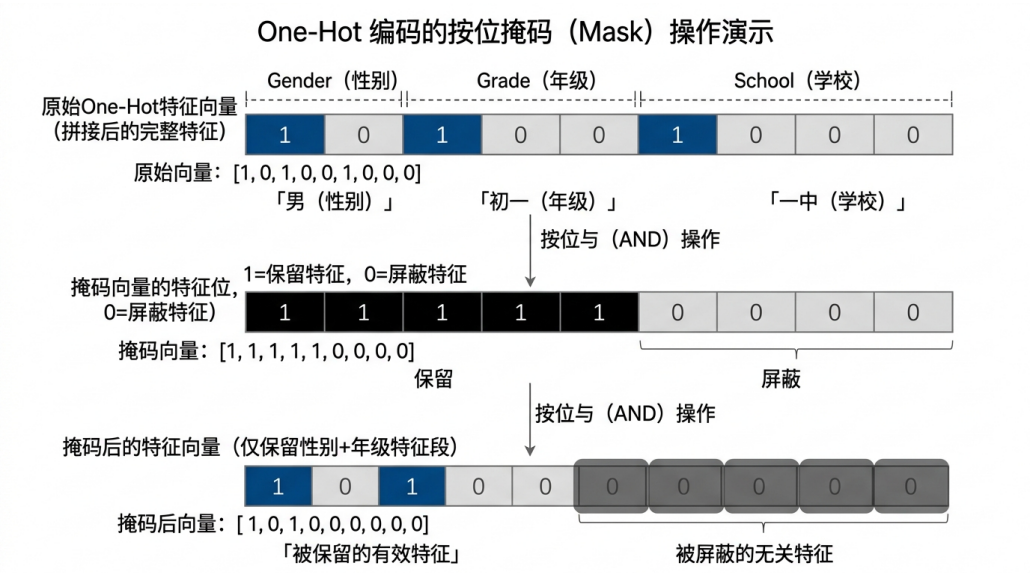

图 5: 按位掩码操作演示

3.3.4 缺点

One-Hot 编码的缺陷在高基数场景中尤为明显：

1. **维度膨胀**：特征规模会随类别数量线性增长，如 100 个城市的特征会直接扩展为 100 维。

2. **数据高度稀疏**：向量中仅一位有效，其余全为 0，信息密度极低，可能导致梯度更新不稳定、过拟合风险上升。

3. **无法表示类别间潜在相似性**：不同类别的向量彼此正交，无法体现业务关联，如“北京”和“上海”的编码距离与“北京”和“拉萨”完全相同。

4 特征变换与规范化

4.1 数据规范化原理

在特征工程流水线中，数据规范化是最基础的环节。不同特征往往带有不同的物理单位和量级，规范化的核心在于消除量纲差异，使多维数据在统一的尺度下进行运算，从而构建起一个公平的特征比较空间。

4.1.1 归一化 (Min-Max Normalization)

归一化（最大-最小规范化）通过线性变换，将数据特征映射到 $[0,1]$ 闭区间。其数学公式为：

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{2}$$

此举的目的是使得各特征对目标变量的影响保持一致，但该操作**会改变特征数据原始的分布形态**。

从统计学稳定性来看，归一化存在一个弱点：**极度缺乏鲁棒性**。其计算强依赖于数据集的两个极端值 $x_{\max}$ 和 $x_{\min}$。在真实的业务数据中，由于传感器故障或录入错误导致的离群点极具破坏性。一旦出现一个极大离群点，分母会瞬间膨胀，导致所有正常样本被压缩在靠近 0 的极小区间内。

4.1.2 Z-Score 标准化 (Standardization)

标准化利用全体样本的均值 μ 与标准差 σ 进行处理，公式为：

$$x^* = \frac{x - \mu}{\sigma} \tag{3}$$

处理后的数据满足均值为 0、方差为 1。标准化变换后，**特征数据的分布形状没有发生改变**。

标准化本质上是坐标轴的平移与等比例缩放，属于线性变换中的保角变换。它更具

鲁棒性，因为 μ 和 σ 综合了全体样本的信息，个别离群点对整体统计指标的拉扯有限。

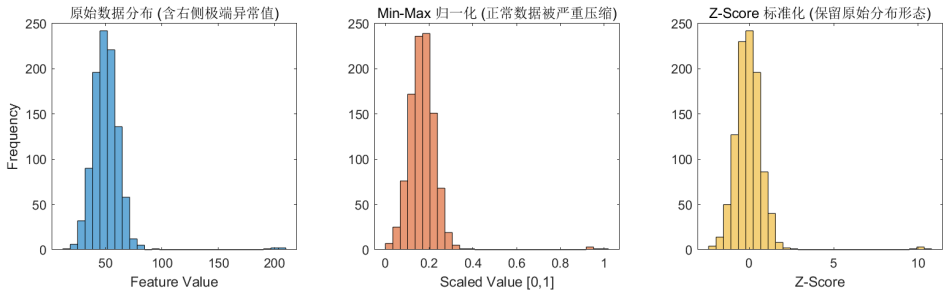

图 6: MATLAB 仿真对比：在包含异常值时，标准化（右）比归一化（中）更好地保留了原始的高斯分布形态。

4.2 数据规范化的模型适用性边界

并非所有算法都需要进行特征缩放。我们需要从算法的数学驱动机制（优化器与距离度量）来判断其适用性。

4.2.1 对量级敏感的模型（必须规范化）

线性模型、基于距离度量的模型（如 KNN 近邻、K-means 聚类）、感知机、SVM 等，通常必须进行数据规范化处理。

为什么量级差异会阻碍梯度下降？从数学上讲，未缩放的特征会导致损失函数的 Hessian 矩阵条件数（Condition Number）过大。反映在几何上，就是损失函数的等高线呈现出极度狭长的椭圆。在这样的地貌中，负梯度方向不再指向全局最优点，模型参数会在“峡谷”两侧剧烈震荡，导致收敛缓慢。

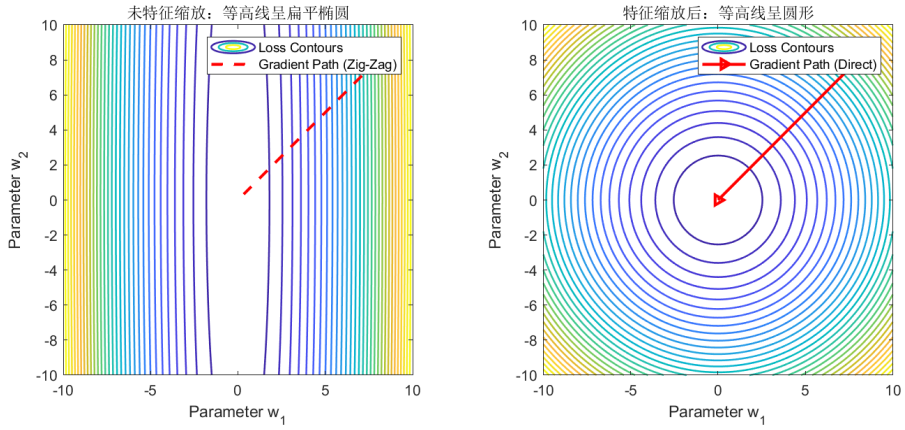

图 7: 等高线仿真：展示了特征缩放如何将“扁平椭圆”修正为“正圆”，使梯度方向直指圆心。

4.2.2 对量级不敏感的模型（无需规范化）

决策树、随机森林、XGBoost、LightGBM 等树模型，一般不需要进行规范化处理。

树模型的分裂点寻找过程是基于特征值的**相对排序**进行的（计算信息增益或基尼系数）。只要变换是单调的（如线性缩放），样本的先后顺序就不会被打乱。因此，无论特征量级是多少，分裂后的子集结果完全一致。

4.3 分箱法（离散化）

4.3.1 核心概念与目的

分箱法（Binning）是将连续型变量离散化为有限个区间（“箱”）的过程。其主要目的包括：

- 增强模型稳定性：离散化后特征对异常值不敏感，减小噪声干扰
- 降低过拟合风险：减少连续值的细微变化对模型的影响
- 引入非线性：使线性模型也能捕捉分段效应，提升表达能力
- 增强可解释性：将数值转化为如“高/中/低”等业务标签，便于理解

4.3.2 常见分箱方法

根据分箱策略的不同，主要分为三类：

1. **等宽分箱（Equal-width Binning）**：将数据范围均分为 k 个等宽区间。适用于数据分布较均匀的场景，但容易受异常值影响导致多数样本挤在少数箱中。
2. **等频分箱（Equal-frequency Binning）**：使每个箱包含大致相同的样本数量。能保证每个箱的代表性，但可能导致不同箱的宽度差异很大。
3. **自定义分箱**：根据业务经验或领域知识手动指定区间边界。例如年龄分段（0-18, 19-35, 36-60, 60+），具有强可解释性。

4.3.3 分箱后处理与编码

分箱后，通常需要将每个箱转换为数值编码：

- **序号编码（Ordinal Encoding）**：为每个箱赋予一个整数顺序，适用于有顺序关系的区间。
- **One-Hot 编码**：将每个箱视为独立类别，避免引入虚假顺序，适合无序分箱。
- **均值编码（Mean Encoding）**：用箱内样本的目标变量均值作为编码值，常用于监督学习。

4.3.4 举例说明

设成绩数据为：`score_list = [63, 64, 88, 71, 42, 60, 99, 70, 32, 88, 34, 69, 83, 52, 66, 92, 82, 58, 66, 41]`

```
# 等宽分箱（自定义边界）
bins = [0, 59, 70, 80, 90, 100]
score_cat = pd.cut(score_list, bins)
print(pd.value_counts(score_cat))
```

输出：

- (59, 70]：7 个
- (0, 59]：6 个
- (80, 90]：4 个
- (90, 100]：2 个
- (70, 80]：1 个

```
# 等频分箱（5个箱，每箱约4个样本）
score_cat = pd.qcut(score_list, 5)
print(pd.value_counts(score_cat))
```

输出：

- (31.999, 50.0]：4 个
- (50.0, 63.6]：4 个
- (63.6, 69.4]：4 个
- (69.4, 84.0]：4 个
- (84.0, 99.0]：4 个

4.3.5 注意事项

- 分箱边界应避免过细或过粗，可通过交叉验证确定最佳箱数
- 对于测试集，需使用训练集的分箱边界进行转换，防止数据泄露
- 异常值可单独作为一个箱，或采用截尾处理

4.4 转换特征构造

4.4.1 基本思想

转换特征构造（Feature Transformation）是指对原始特征进行数学函数映射或组合，生成具有更强表达能力的新特征。其目标是：

- 改善特征分布（如使用对数变换减轻长尾分布）
- 揭示隐藏关系（如通过乘积、比值构造交互特征）
- 引入领域知识（如根据单价和销量计算销售额）

4.4.2 常见转换方式

1. **单变量数学变换：**对数、指数、平方根、Box-Cox 变换等，用于处理偏态分布或稳定方差。
2. **多项式特征（Polynomial Features）：**生成特征的平方、乘积等交叉项，例如 x_1^2, x_1x_2, x_2^2，可使线性模型拟合非线性关系。
3. **组合特征（Arithmetic Combinations）：**通过加、减、乘、除构造业务相关指标，如“人均消费 = 总金额/订单数”、“毛利率 = (售价-成本)/售价”。
4. **比率与差值：**例如“距离” = 目的地经度 - 起点经度，“增长率” = (本月销量 - 上月销量)/上月销量。
5. **分箱后变换：**将连续变量分箱后再进行统计编码（如均值编码）。

4.4.3 举例说明

1. 销售额 = 单价 × 销售量（业务组合特征）
2. 利润 = 售价 - 原价（差值特征）
3. 环比增长率 = (本月销售额 - 上月销售额) / 上月销售额（比率特征）

4. 对长尾分布的特征 x 做 $\log(1+x)$ 变换（分布调整）

5. 使用多项式特征：若原始特征为 x_1, x_2，加入 x_1^2, x_2^2, x_1x_2 可拟合二次曲面。

4.4.4 实践建议

- 特征转换应与模型类型匹配：树模型对单调变换不敏感，而线性模型依赖变换实现非线性。
- 避免引入高度相关的冗余特征（如同时保留 x 和 x^2 可能造成多重共线性）。
- 对于比率特征，注意分母为零的情况，可加小常数 ϵ 平滑。

4.5 特征变换的深层原理

4.5.1 为什么要进行特征变换

除了消除量纲影响外，特征变换具有以下深层次的工程意义：

1. **优化条件数与收敛速度**：特征缩放能改善损失函数曲面地貌，减小 Hessian 矩阵的条件数，使梯度下降法能以更平滑、更直接的路径逼近全局最优点。
2. **数值稳定性**：在深度学习中，未经变换的输入会导致神经元在训练初期就进入激活函数的饱和区，引发梯度消失。
3. **正则化的公平性**：如果特征量级不同，L2 正则化会更多地惩罚那些原始量级大的特征权重。特征变换确保了正则化惩罚在不同维度特征之间是公平的。
4. **满足统计假设**：如 Log 变换能将异方差性转变为同方差性，满足线性回归等模型对输入分布的预设。

4.5.2 什么时候需要做，什么时候不需要做

其必要性完全取决于模型底层的数学计算方式：

- **需要做的情况：**基于“度量”或“权重加和”的模型（KNN、SVM、神经网络、K-means 等）。如果不缩放，小量级特征的波动会被大数据值特征直接“淹没”。
- **不需要做的情况：**基于“排序”或“概率密度”的模型（决策树、XGBoost、随机森林、朴素贝叶斯）。由于切分只关心数值大小的排序关系，单调变换对模型分裂逻辑无影响。

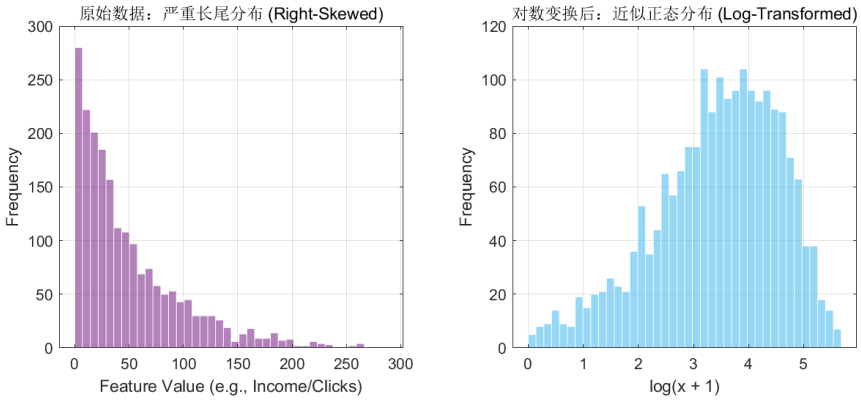

图 8: MATLAB 仿真：对数变换对右偏态（长尾）数据的分布重塑效果展示。

5 聚合特征构造

5.1 定义

聚合特征构造（Aggregation Feature Construction）是特征工程中的一种重要技术，主要通过对多个特征进行分组聚合来实现新特征的构造。这些特征通常来自同一张数据表，或者来自多张表通过主键联立后的结果。

核心思想是利用数据集中天然存在的**一对多（one-to-many）**关联关系，对观测值进行分组，然后在每个分组上计算各类统计量，从而将细粒度的信息汇总为粗粒度的特征。常见分组统计量包括：

- **集中趋势统计量**：算术平均数（Mean）、中位数（Median）、众数（Mode）
- **离散程度统计量**：标准差（Standard Deviation）、方差（Variance）
- **极值统计量**：最小值（Minimum）、最大值（Maximum）
- **频数统计量**：计数（Count）、去重计数（Distinct Count）

通过聚合特征构造，可以有效地将高基数、细粒度的原始数据转化为对模型预测更有信息量的低维度特征，同时降低数据的噪声水平。

5.2 核心数学表达

设主实体为 x_i，与之关联的明细集合为 $S_i = \{v_{i1}, v_{i2}, \ldots, v_{in_i}\}$，通过聚合函数 $g(\cdot)$ 构造新特征 $f_i = g(S_i)$。常用统计量：

- 均值：$\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} v_{ij}$
- 最大值：$\max(S_i)$，最小值：$\min(S_i)$
- 方差：$\sigma_i^2 = \frac{1}{n_i} \sum_{j=1}^{n_i} (v_{ij} - \mu_i)^2$，标准差：$\sigma_i$
- 频数：n_i

- 分位数、极差、变异系数等

5.3 时间窗口聚合

在实际业务中，越新的行为信息越有预测价值，常按时间窗口构造多套聚合特征：

$$S_i^{(w)} = \{v_{ij} \mid t - w < t_{ij} \le t\}, \quad f_i^{(w)} = g\left(S_i^{(w)}\right)$$

例如同时构造过去 7 天、30 天、90 天的均值与频数。

5.4 举例说明：电商场景

场景：用户表与订单表通过 `user_id` 关联，目标预测用户是否购买高价值商品。

表 1: 订单表原始数据示例

order_id	user_id	amount	order_date
1001	U001	350.00	2025-03-01
1002	U001	120.50	2025-03-15
1003	U002	89.00	2025-03-02
1004	U001	560.00	2025-04-01
1005	U003	230.00	2025-03-20

聚合特征构造结果：

表 2: 聚合特征构造结果

user_id	avg_amount	max_amount	min_amount	std_amount	order_count
U001	343.50	560.00	120.50	179.73	3
U002	89.00	89.00	89.00	0.00	1
U003	230.00	230.00	230.00	0.00	1

结果分析：

- U001 的 `avg_amount` 为 343.50，`std_amount` 较高（179.73），消费波动大，可能属于冲动型消费者。
- U002 和 U003 仅有一次购买记录，标准差为 0，属于新用户或低频用户。

这些聚合特征可直接用于机器学习模型训练，帮助理解用户消费模式，提升预测精度。

5.5 实战流程与注意事项

1. 确定主实体与分组键（如 `user_id`）
2. 确定聚合窗口（全历史或滑动窗口）
3. 按字段类型设计聚合函数（数值用均值/方差，类别用占比/去重计数）
4. 统一命名并拼接回主表

注意事项：

- 避免未来信息泄露
- 对缺失分组补零或全局统计量
- 注意异常值影响
- 高基数类别优先使用占比或目标编码

6 特征提取与特征选择

6.1 特征工程与降维问题的背景

在机器学习和模式识别任务中，**原始数据往往包含大量冗余信息、噪声和无关特征**。直接使用高维特征训练模型，会带来两个致命问题：

1. **计算复杂度激增**：特征维度越高，模型的训练时间、内存占用和推理成本都会显著上升，甚至无法在可接受时间内完成训练。
2. **维数灾难（Curse of Dimensionality）**：随着特征维度增加，数据在高维空间中会变得极度稀疏。模型对样本量的需求呈指数级增长，泛化能力反而下降，极易发生过拟合。

为了解决这些问题，我们通常会采用**降维技术**，其核心目标是：

> 在尽可能保留数据核心信息的前提下，降低特征维度，去除冗余与噪声，提升模型效率与泛化能力。

在众多降维方法中，**特征提取（Feature Extraction）**与**特征选择（Feature Selection）**是两种最核心、最基础的手段。它们的目标一致，但实现路径、技术理念和适用场景存在本质差异。

6.2 特征提取与特征选择的核心对比

表 3: 特征提取与特征选择对比

对比维度	特征提取	特征选择
共同点	都从原始特征中找出最有效的特征；都能帮助减少特征的维度、数据冗余。	
区别	强调通过特征转换得到一组具有明显物理或统计意义的特征。	从特征集合中挑选一组具有明显物理或统计意义的特征子集。
是否改变特征空间	是，生成全新特征	否，仅保留子集
可解释性	新特征往往失去原始物理意义	完全保留原始物理含义，可解释性强
计算复杂度	变换过程计算量较大	搜索最优子集是 NP-hard 问题，依赖启发式策略
典型应用场景	图像降维、信号处理、文本语义表示	生物信息学、金融风控、医疗数据
代表性方法	PCA、LDA、ICA、SIFT、HOG、Word2Vec	Filter、Wrapper、Embedded（L1 正则化）
优点	发现潜在结构，降维彻底	可解释性好，训练速度快
缺点	缺乏物理意义，可能丢失信息	搜索空间大，易遗漏组合效应

6.3 特征提取：从原始数据到有效表示

6.3.1 基本概念

特征提取的处理对象是**原始数据**，即未经特征工程处理的原始形态数据，通常在特征选择之前进行。其核心目的是：

> 自动地构建新的特征，将原始数据转换为一组具有明显物理意义或统计意义的新特征表示。

简单来说，特征提取就是“用新的视角看数据”：它不是简单地挑选旧特征，而是通过数学变换，把原始数据的关键信息“压缩”到一个更低维的空间里。

6.3.2 常用特征提取方法分类

- **降维分析类：**PCA（无监督）、LDA（有监督）、ICA（盲源分离）

- **图像处理类**：SIFT、HOG、Gabor 滤波器
- **文本挖掘类**：词袋模型、Word2Vec、GloVe

6.4 特征选择：从给定特征集合中筛选最优子集

6.4.1 基本概念

特征选择是指从给定的特征集合中按照某种评价准则选出一个相关特征子集的过程。主要动机：

1. 解决维数灾难问题
2. 降低学习任务难度
3. 简化模型，提升效率

6.4.2 特征的分类

- **相关特征**：对学习任务有用，必须保留
- **无关特征**：与目标变量无统计依赖，应剔除

6.4.3 特征选择的三大策略

1. **Filter（过滤式）**：用统计指标评估相关性，高效但忽略特征交互
2. **Wrapper（包裹式）**：以学习器性能为评价准则，准确但计算开销大
3. **Embedded（嵌入式）**：将选择融入训练（如 L1 正则化、决策树重要性），兼顾效率与效果

6.5 模型性能与计算复杂度的权衡

特征子集规模的确定本质上是多目标优化问题。

表 4: 特征数量对模型的影响

评估维度	保留较多特征	保留较少特征
模型性能	信息完整，但可能引入噪声	可能略有下降，但噪声更少
计算复杂度	显著增加	效率高
过拟合风险	极易过拟合	泛化能力强
可解释性	复杂，难以理解	精简，易于解释
存储与传输开销	高	低

理想策略遵循奥卡姆剃刀原则：在泛化性能无显著损失的前提下，尽可能压缩特征集规模。

6.6 总结与工程建议

- 优先顺序：先特征提取（如 PCA），再进行特征选择
- 可解释性优先场景（金融风控、医疗诊断）：优先特征选择
- 性能优先场景（图像识别、文本分类）：优先特征提取
- 没有“最好”的方法，只有“最合适”的方法