5



数据预处理：
数据归一化与标准化

授课教师：[李阳源]

SL02:今日学习目标



- 理解数据归一化与标准化的概念
- 了解为什么需要归一化和标准化
- 学习主要的归一化/标准化技术
- Min-Max归一化
- Z-score标准化
- 对数变换
- 了解不同情况下如何选择缩放方法

SL03: 数据缩放的必要性

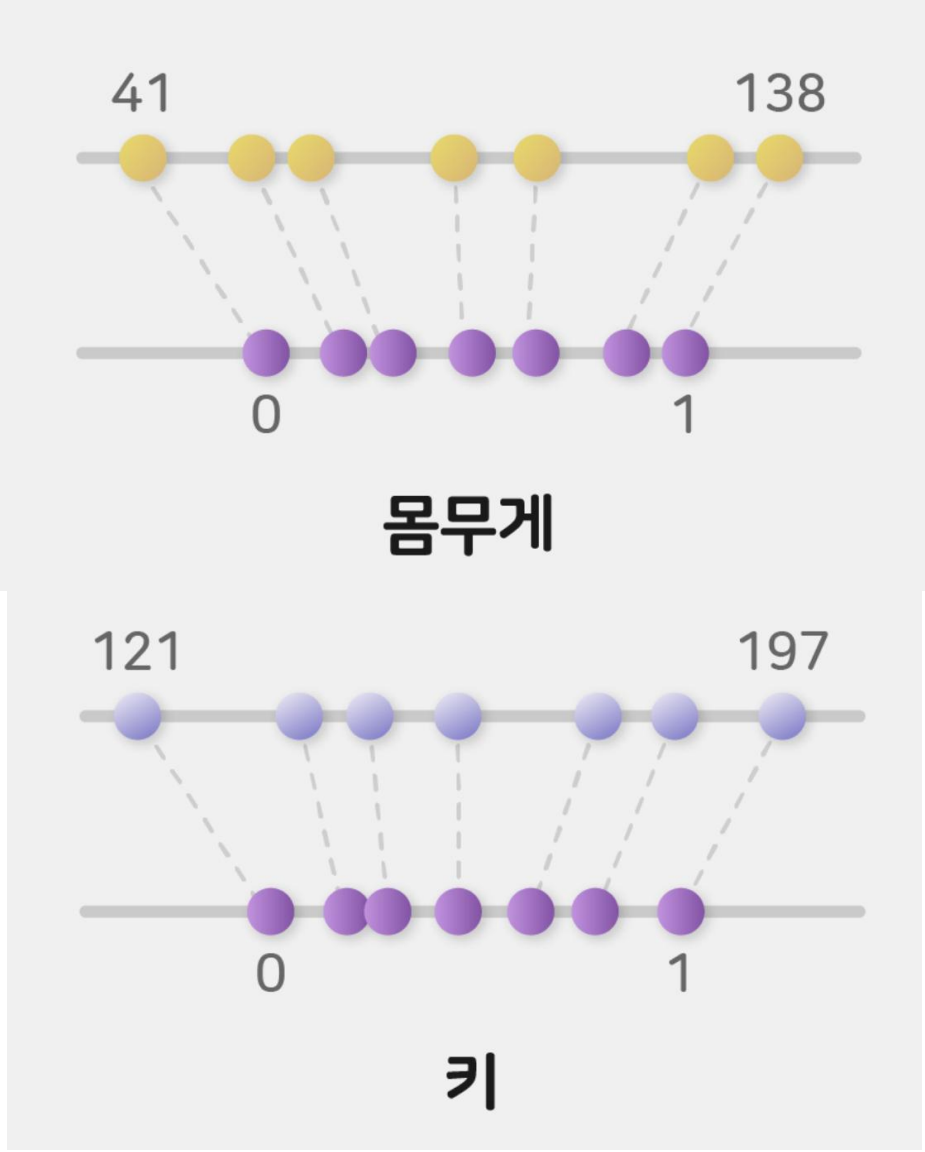


- 当不同特征的尺度差异很大时出现的问题
- 例子：收入(几百万)vs年龄(几十)
- 机器学习算法可能会给予尺度较大的特征更多权重
- 主要影响：
  - ◆对基于距离的算法影响很大(KNN、K-means等)
  - ◆影响基于梯度下降的算法的收敛速度
  - ◆影响正则化项

SL04: 需要缩放的情况



- 数据特征范围差异很大的情况
- 特征之间单位不同的情况(米、千克、元等)
- 存在较多离群值的情况
- 使用假设正态分布的算法时
- 使用基于梯度的优化算法时



SL05: 数据归一化(Normalization)



- 将数据转换到特定范围(通常为0~1)的过程
- 使所有特征具有相同的尺度
- 在神经网络中经常使用
- 保持特征之间的相对关系

SL06: Min-Max归一化



- 最基本的归一化方法
- 将所有数据转换为0~1之间的值
- 计算公式：

  X_normalized = (X - X_min) / (X_max - X_min)

- 转换到用户定义范围[a, b]时：

  X_normalized = a + (X - X_min) * (b - a) / (X_max - X_min)

SL07: Min-Max归一化示例



- 原始数据: [5, 10, 15, 20, 25]
- X_min = 5, X_max = 25
- 归一化计算:
  - (5-5)/(25-5) = 0
  - (10-5)/(25-5) = 0.25
  - (15-5)/(25-5) = 0.5
  - (20-5)/(25-5) = 0.75
  - (25-5)/(25-5) = 1
- 归一化结果: [0, 0.25, 0.5, 0.75, 1]

SL08: Min-Max归一化的特点



- 优点:
  - 直观比较
  - 所有特征具有相同尺度
  - 计算简单
- 缺点:
  - 对离群值敏感
  - 可能不适合假设正态分布的算法
- 适用场景:
  - 图像处理
  - 神经网络输入

SL09: 数据标准化(Standardization)



- 将数据转换为均值为0、标准差为1的过程
- 也称为Z-score标准化
- 在数据呈正态分布时效果好
- 对离群值较不敏感

SL10: Z-score标准化



- 利用数据的均值和标准差
- 计算公式:
  - X_standardized = (X - μ) / σ
  - μ: 均值
  - σ: 标准差
- 结果: 均值为0、标准差为1的分布

SL11: Z-score标准化示例



- 原始数据: [5, 10, 15, 20, 25]
- 均值(μ) = 15, 标准差(σ) = 7.07
- 标准化计算:
  - (5-15)/7.07 = -1.41
  - (10-15)/7.07 = -0.71
  - (15-15)/7.07 = 0
  - (20-15)/7.07 = 0.71
  - (25-15)/7.07 = 1.41
- 标准化结果: [-1.41, -0.71, 0, 0.71, 1.41]

SL12: Z-score 标准化的特点



- 优点:
  - 对离群值较不敏感
  - 适合假设正态分布的算法
  - 线性变换，保留数据特性
- 缺点:
  - 没有特定的范围限制
  - 比Min-Max更难直观解释
- 适用场景:
  - 线性回归
  - SVM(支持向量机)
  - 主成分分析(PCA)

SL13: Min-Max归一化 vs Z-score标准化



1. ![归一化与标准化比较](示例图片: 同一数据集的归一化与标准化比较)

特性	Min-Max归一化	Z-score标准化
结果范围	0 ~ 1 (或指定范围)	无限制 (通常在 -3 ~ +3)
对离群值敏感度	高	低
分布假设	无	假设正态分布
主要用途	神经网络、图像处理	线性回归、PCA、SVM

SL14: 对数变换(Log Transformation)



- 非线性变换方法
- 将偏斜分布变换为接近正态分布
- 使指数关系线性化
- 减少离群值影响
- 计算公式:

$$X_log = \log(X)$$

- 注意: X > 0 (通常使用 X+1)

SL15: 对数变换示例



- ![对数变换前后](示例图片: 对数变换前后分布比较)
- 原始数据: [1, 10, 100, 1000, 10000]
- 对数变换(底数10):

  log10(1) = 0

  log10(10) = 1

  log10(100) = 2

  log10(1000) = 3

  log10(10000) = 4

- 变换结果: [0, 1, 2, 3, 4]

SL16: 对数变换的特点



- 优点:
  - 正态化偏斜分布
  - 线性化指数关系
  - 减少离群值影响
- 缺点:
  - 不能直接应用于0或负值
  - 数据解释可能不够直观
- 适用场景:
  - 收入、人口等幂律分布数据
  - 具有较大正偏度的数据

SL17: 其他变换方法



- Robust Scaling (鲁棒缩放)
  - 使用中位数和四分位范围(IQR)
  - 对离群值非常鲁棒
  - X_robust = (X - median) / IQR
- MaxAbs Scaling (绝对值最大缩放)
  - 除以最大绝对值
  - 对稀疏数据有用
  - X_maxabs = X / |X|_max
- Quantile Transformation (分位数变换)
  - 转换为均匀分布或正态分布
  - 改变分布形态本身

SL18: 如何选择合适的缩放方法？



情况	推荐方法
神经网络、图像数据	Min-Max归一化
线性模型、假设正态分布	Z-score标准化
存在大量离群值	Robust Scaling
偏斜分布	对数变换
稀疏矩阵	MaxAbs Scaling
使用非参数方法时	Quantile Transformation

SL19: 缩放时的注意事项



- 训练/测试集分离后应用缩放
  - 防止测试数据信息泄露到训练中
- 在测试集上使用训练集的统计量
  - 使用训练数据的均值、标准差等
- 处理离群值后再缩放
  - 离群值会影响缩放范围
- 利用领域知识
  - 并非所有特征都需要相同的缩放

SL20: 缩放对机器学习性能的影响



![缩放影响](示例图片: 缩放前后算法性能比较)

- KNN(K-最近邻): 显著提高
- 线性回归: 只改变系数规模，结果相同
- 决策树: 几乎无影响
- SVM、神经网络: 显著提高
- 逻辑回归: 提高，特别是使用正则化项时

SL21: 实例：房价预测



预测公寓价格时:

- 面积: 20~200m² (变动范围小)
- 价格: 1亿~30亿韩元 (变动范围大)
- 建筑年份: 1980~2023 (变动范围中等)

不使用缩放时:

- 价格特征主导模型
- 面积、建筑年份的重要性可能被忽视

应用缩放后:

- 所有特征均等考虑
- 可能更准确预测

SL22: 实例：人脸识别



在人脸识别算法中:

- 像素值: 0~255
- 各种光照条件
- 相机设置差异

应用Min-Max归一化:

- 将所有图像转换为相同亮度范围
- 补偿光照条件差异
- 提高特征提取性能

SL23: 理解度测试



1. 以下哪种缩放方法对离群值最不敏感？

a) Min-Max归一化 b) Z-score标准化 c) Robust Scaling d) 对数变换

2. 对于收入数据等具有较大正偏度的数据，最适合的变换方法是？

a) Min-Max归一化 b) Z-score标准化 c) 对数变换 d) MaxAbs Scaling

3. 数据为[-100, 200, 300, 400, 500]时，Min-Max归一化后结果是?

a) [0, 0.5, 0.67, 0.83, 1] b) [0, 0.25, 0.5, 0.75, 1] c) [-1, 0, 0.33, 0.67, 1] d) [-0.5, 0, 0.17, 0.33, 0.5]

SL24: 理解度测试答案



1. c) Robust Scaling
   - 使用中位数和IQR，因此对离群值最不敏感
2. c) 对数变换
   - 将具有正偏度的数据转换为接近正态分布
3. a) [0, 0.5, 0.67, 0.83, 1]
   - 计算: (X - X_min) / (X_max - X_min)
   - (-100-(-100))/(500-(-100)) = 0
   - (200-(-100))/(500-(-100)) = 0.5
   - (300-(-100))/(500-(-100)) = 0.67
   - (400-(-100))/(500-(-100)) = 0.83
   - (500-(-100))/(500-(-100)) = 1

SL25: 总结与下节课预告



今日学习内容

- 数据归一化与标准化的概念
- Min-Max归一化方法及特点
- Z-score标准化方法及特点
- 对数变换及其他缩放方法
- 根据情况选择合适的缩放方法

下节课预告

- 下一主题: 数据编码(Encoding)
- 分类变量处理方法
- 标签编码、独热编码等
- 多类别处理方法

提问环节

1. 有问题吗?

PL01: 数据缩放与归一化实践



1. 第5周实践课

PL02: 今日学习目标



- 学习使用Python进行数据缩放的方法
- 掌握scikit-learn (科学学习库) 预处理模块的使用方法
- 实际实现各种缩放技术
  - Min-Max归一化
  - Z-score标准化
  - 对数变换
- 观察缩放对机器学习性能的影响

PL03: 开发环境设置



- 使用的IDE: PyCharm (派尔参)
- 所需库:
  - NumPy (数字派): 数值计算
  - pandas (熊猫): 数据处理
  - scikit-learn (科学学习): 预处理工具
  - matplotlib (数学绘图库): 可视化
  - seaborn (海洋图): 高级可视化

PL04: 项目设置(통과)



- 创建新项目:
- 创建文件:
  - scaling_basics.py: 基本缩放实践
  - scaling_visualization.py: 缩放可视化
  - scaling_ml_impact.py: 对机器学习的影响
- 虚拟环境设置 (可选)
  - 设置项目解释器
  - 安装所需包: pip install numpy pandas scikit-learn matplotlib seaborn

PL05: 导入所需库



PL06: 创建样本数据(generation sample data)



原始数据统计：

	年龄	年薪	工作时间
count	100.000000	1.000000e+02	100.000000
mean	38.790000	6.804445e+06	53.420000
std	11.556008	2.019672e+06	15.010555
min	20.000000	3.039353e+06	30.000000
25%	28.000000	5.122741e+06	40.000000
50%	39.500000	6.949358e+06	53.500000
75%	47.250000	8.624246e+06	65.250000
max	59.000000	9.914550e+06	79.000000

PL07: Min-Max归一化实践



原始和归一化数据样本：

	年龄	年薪	工作时间	年龄_归一化	年薪_归一化	工作时间_归一化
0	58	7545583	41	0.974359	0.655433	0.224490
1	48	4870230	68	0.717949	0.266302	0.775510
2	34	6747389	31	0.358974	0.539335	0.020408
3	27	7142807	32	0.179487	0.596849	0.040816
4	40	7683874	78	0.512821	0.675547	0.979592

PL08: Z-score标准化实践



```
Z-score标准化结果:
              年龄_标准化         年薪_标准化       工作时间_标准化
count   1.000000e+02  1.000000e+02  1.000000e+02
mean    8.215650e-17  2.020606e-16 -1.124101e-16
std     1.005038e+00  1.005038e+00  1.005038e+00
min    -1.634186e+00 -1.873601e+00 -1.568096e+00
25%    -9.384173e-01 -8.368568e-01 -8.985415e-01
50%     6.174943e-02  7.211207e-02  5.356432e-03
75%     7.357749e-01  9.055773e-01  7.920825e-01
max     1.757684e+00  1.547664e+00  1.712719e+00
```

PL09: 对数变换实践



原始和对数变换数据样本：

	年龄	年薪	工作时间	年龄_对数	年薪_对数	工作时间_对数
0	58	7545583	41	4.060443	15.836473	3.713572
1	48	4870230	68	3.871201	15.398652	4.219508
2	34	6747389	31	3.526361	15.724666	3.433987
3	27	7142807	32	3.295837	15.781616	3.465736
4	40	7683874	78	3.688879	15.854634	4.356709

PL10: Robust缩放和MaxAbs缩放

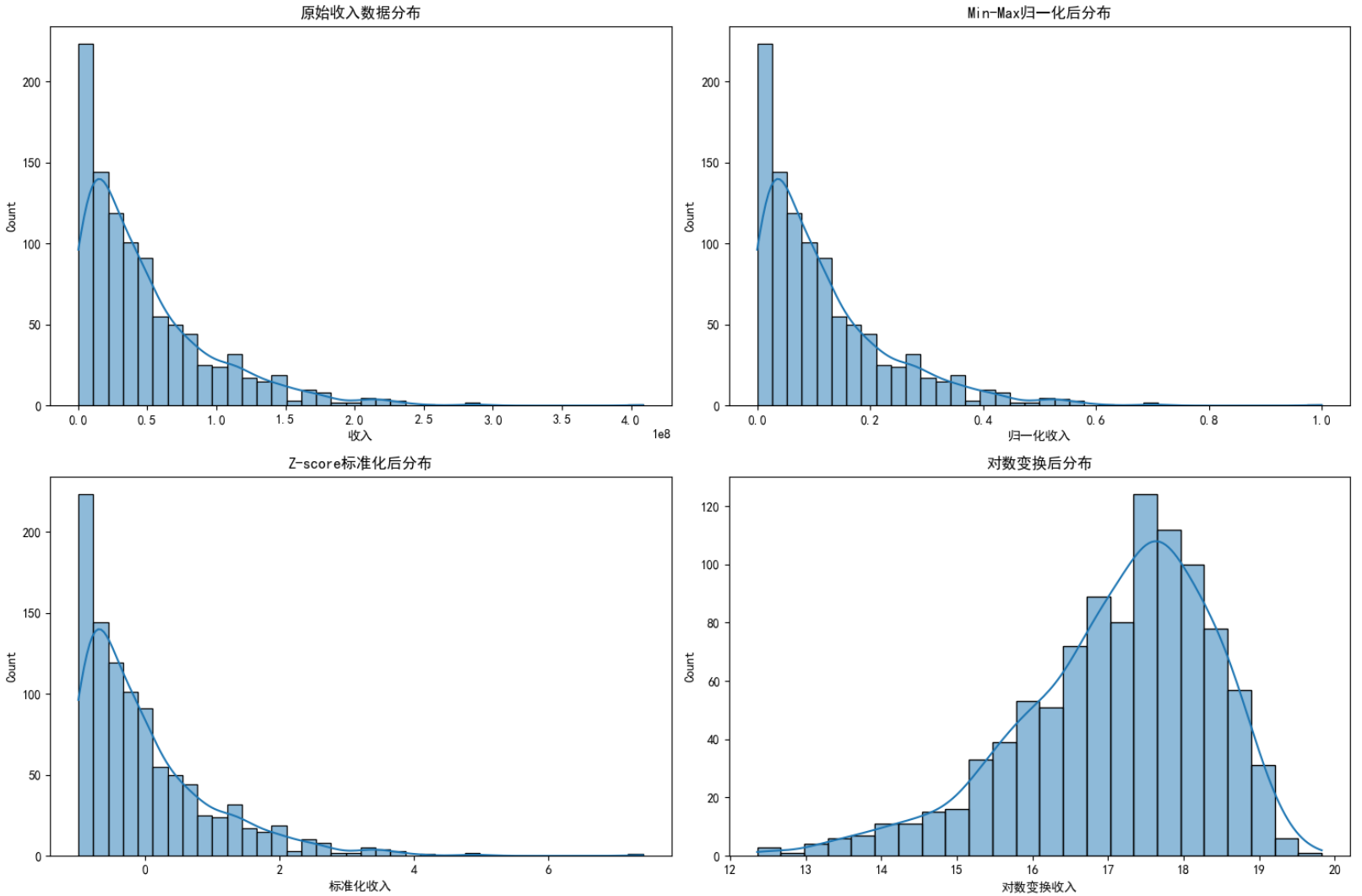


```
Robust缩放结果:
            年龄_鲁棒        年薪_鲁棒      工作时间_鲁棒
count  100.000000  100.000000  100.000000
mean    -0.036883   -0.041386   -0.003168
std      0.600312    0.576801    0.594477
min     -1.012987   -1.116664   -0.930693
25%     -0.597403   -0.521666   -0.534653
50%      0.000000    0.000000    0.000000
75%      0.402597    0.478334    0.465347
max      1.012987    0.846834    1.009901
```

```
MaxAbs缩放结果:
          年龄_绝对最大      年薪_绝对最大    工作时间_绝对最大
count  100.000000  100.000000  100.000000
mean     0.657458    0.686309    0.676203
std      0.195865    0.203708    0.190007
min      0.338983    0.306555    0.379747
25%      0.474576    0.516689    0.506329
50%      0.669492    0.700925    0.677215
75%      0.800847    0.869858    0.825949
max      1.000000    1.000000    1.000000
回车键：缩放可视化 - 分布比较
```

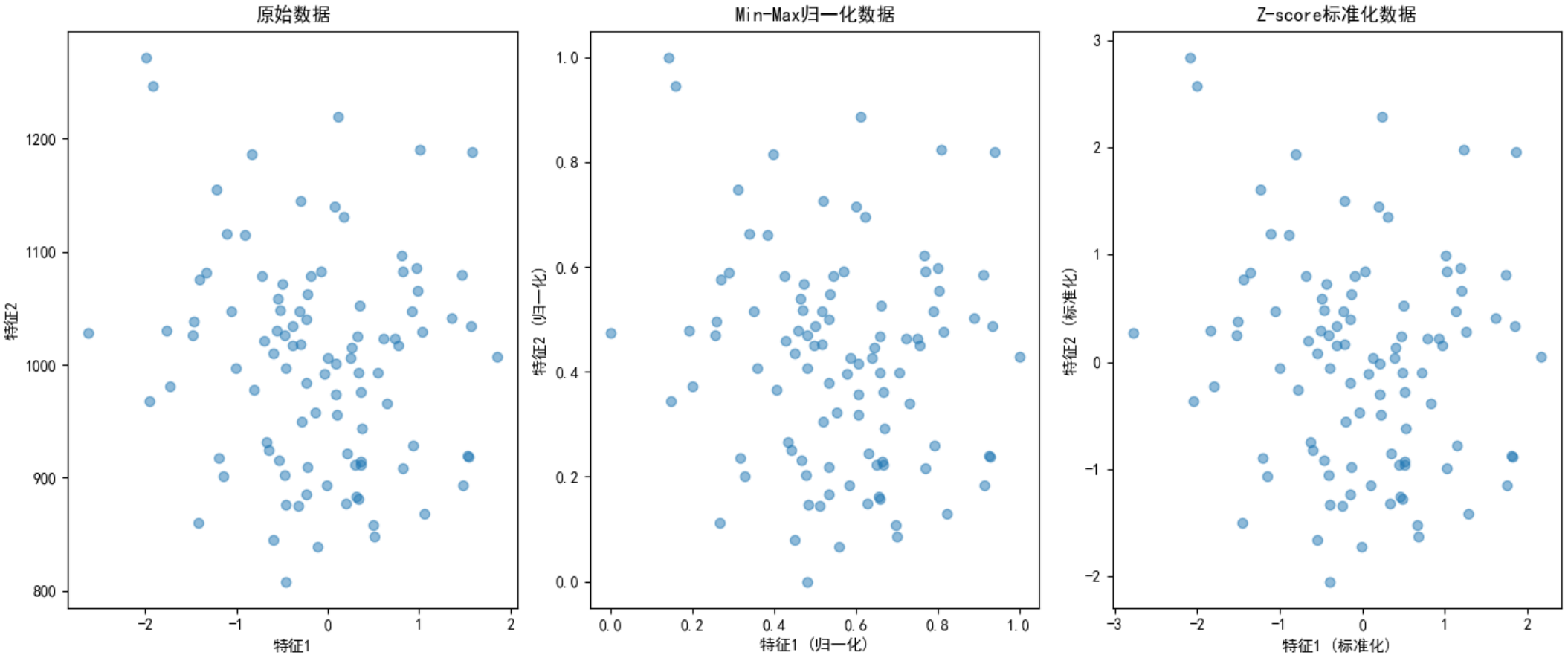
PL11: 缩放可视化 - 分布比较





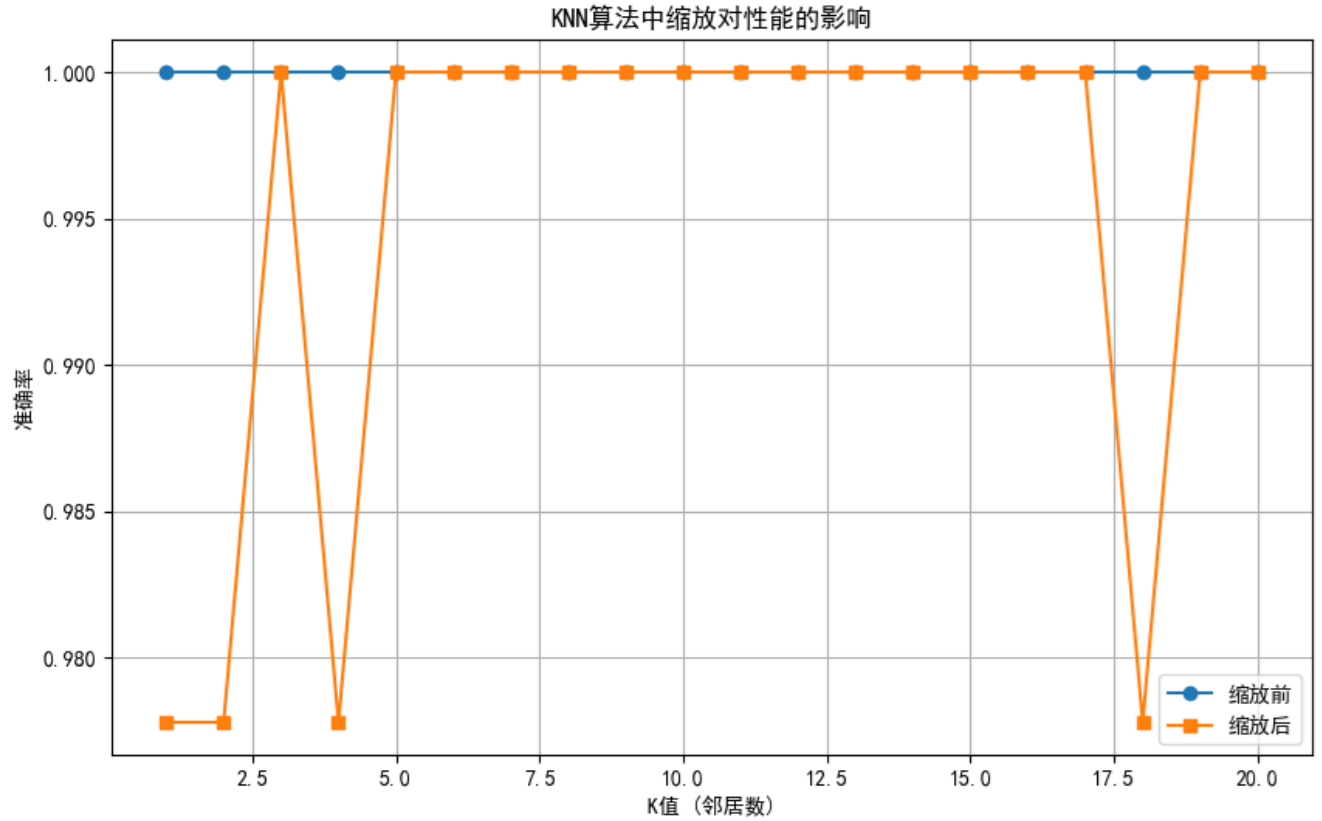
PL12: 缩放可视化 - 视觉比较





PL13: 缩放对机器学习的影响

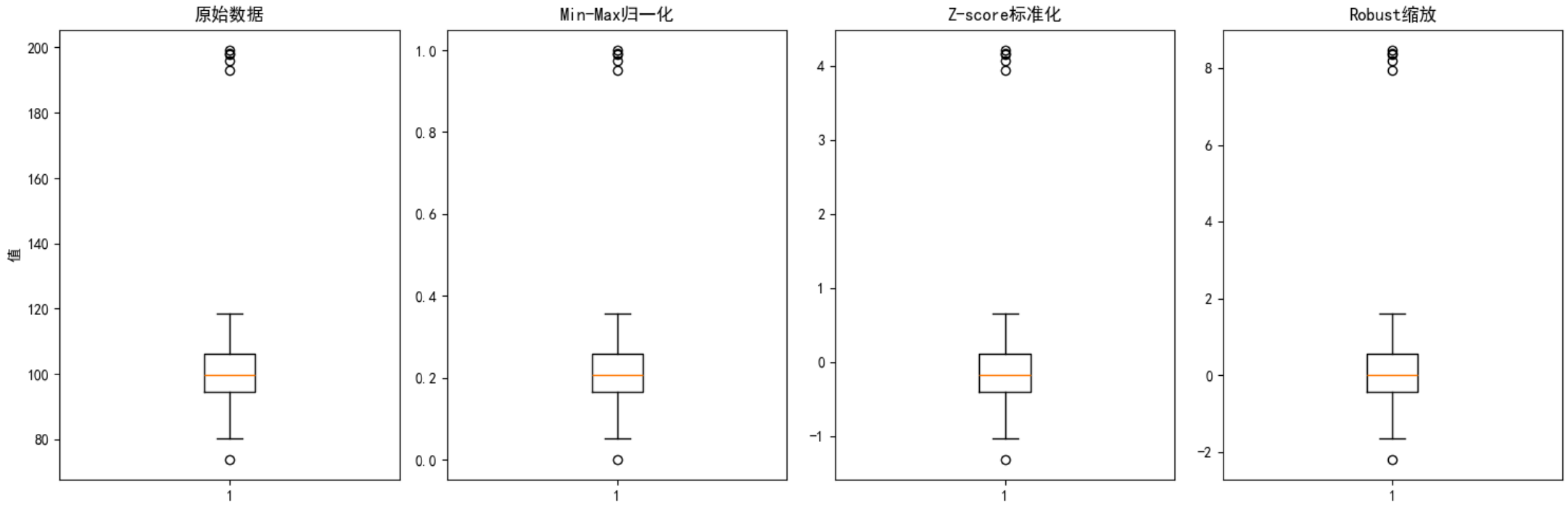





```
缩放前平均准确率: 1.0
缩放后平均准确率: 0.9955555555555555
```

PL14: 有异常值情况下的缩放比较





PL15: 实际数据集缩放实践

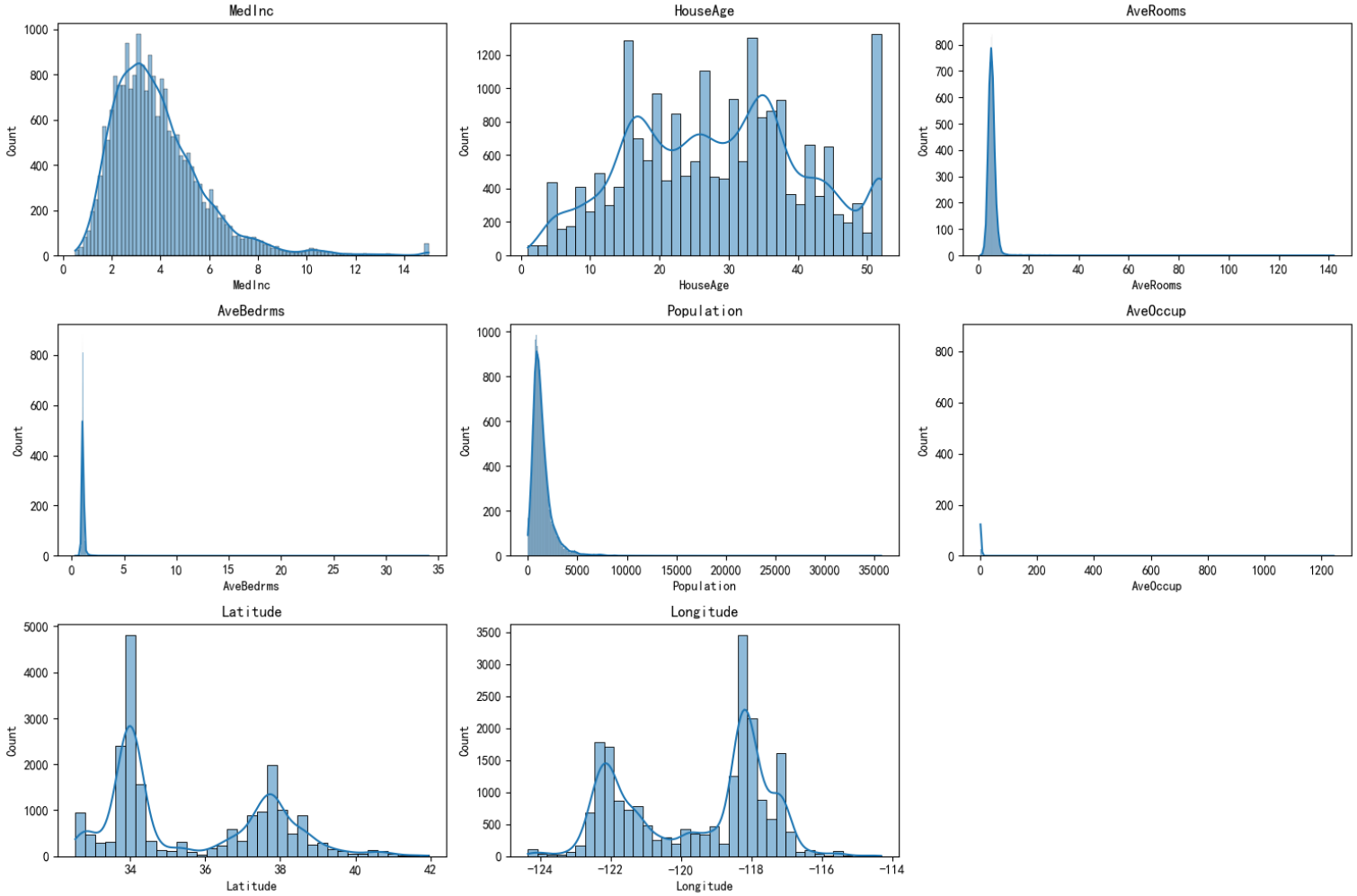


原始数据统计：

	MedInc	HouseAge	...	Latitude	Longitude
count	20640.000000	20640.000000	...	20640.000000	20640.000000
mean	3.870671	28.639486	...	35.631861	-119.569704
std	1.899822	12.585558	...	2.135952	2.003532
min	0.499900	1.000000	...	32.540000	-124.350000
25%	2.563400	18.000000	...	33.930000	-121.800000
50%	3.534800	29.000000	...	34.260000	-118.490000
75%	4.743250	37.000000	...	37.710000	-118.010000
max	15.000100	52.000000	...	41.950000	-114.310000

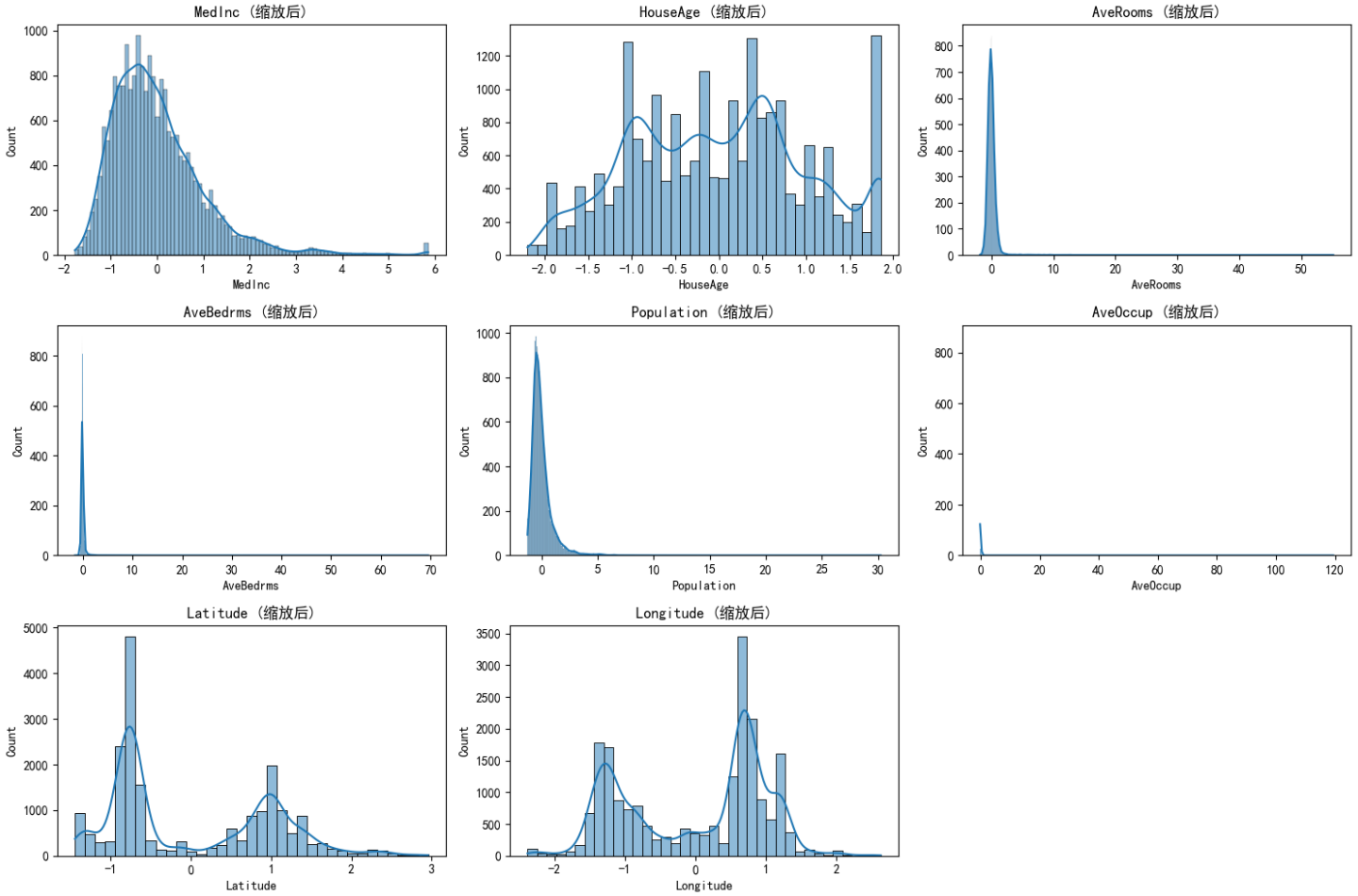
PL15: 实际数据集缩放实践





PL15: 实际数据集缩放实践

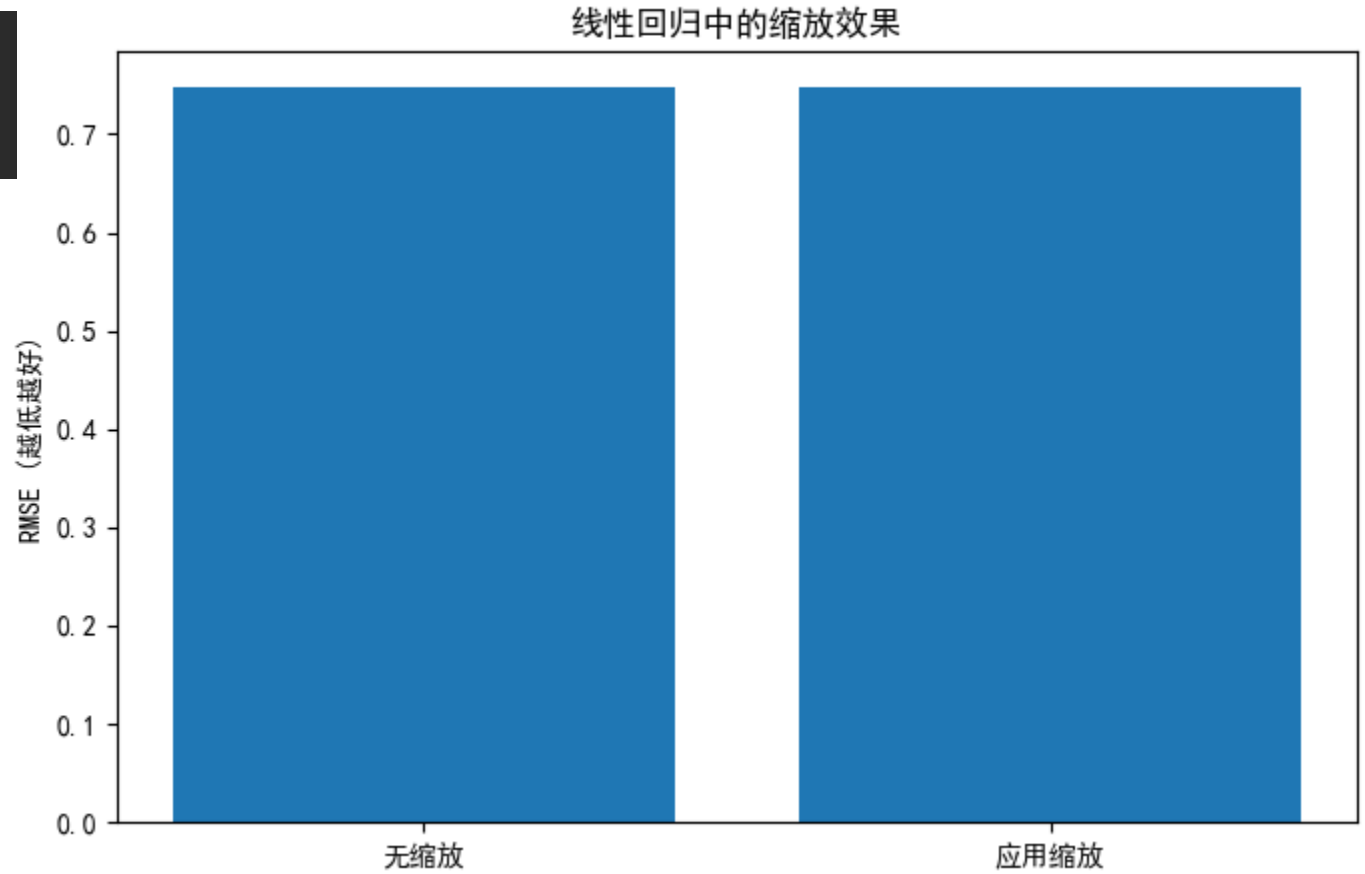




PL16: 自动化缩放管道

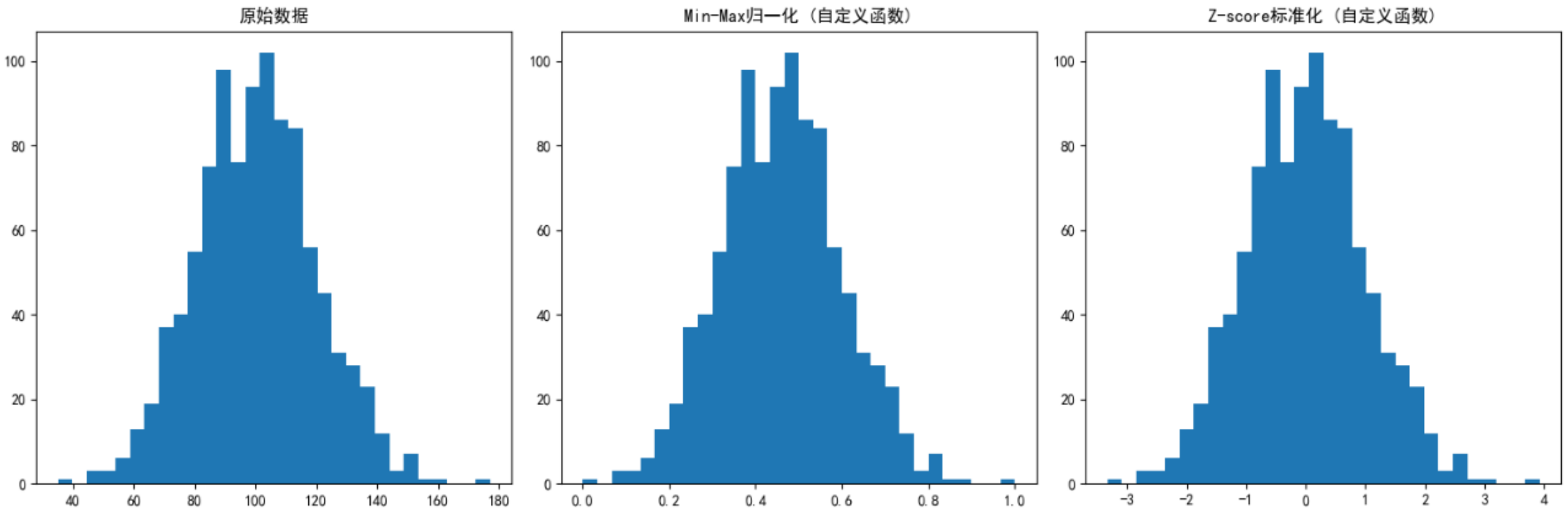


```
回车键：自动化缩放管道无缩放RMSE: 0.7472
应用缩放RMSE: 0.7472
```



PL17: 创建自定义数据缩放函数
(Create custom data scaling function)





PL18: 理解度测试



1. 以下哪项是应用缩放的正确顺序?

a) 对整个数据集缩放 → 分割训练/测试集
b) 分割训练/测试集 → 用训练数据拟合缩放器 → 对训练和测试数据应用缩放
c) 分割训练/测试集 → 分别独立缩放
d) 应用缩放 → 移除异常值 → 分割训练/测试集

2. 以下Python代码的执行结果是什么?

```
from sklearn.preprocessing import MinMaxScaler
import numpy as np

data = np.array([[10], [20], [30], [40], [50]])
scaler = MinMaxScaler()
scaled = scaler.fit_transform(data)
print(scaled[2][0])
```

a) 0.5 b) 30 c) 0.75 d) 0.25

PL18: 理解度测试



1. 在KNN(K-最近邻)算法中，缩放为什么重要?

a) 提高计算速度

b) 使特征选择更容易

c) 因为它是基于距离的算法，对特征尺度敏感

d) 使其对异常值不那么敏感

PL19: 理解度测试答案



1. b) 分割训练/测试集 → 用训练数据拟合缩放器 → 对训练和测试数据应用缩放
   防止测试数据信息泄露到训练过程中
   需要用相同的缩放器转换所有数据
2. a) 0.5
   MinMaxScaler将数据变换到[0, 1]范围
   (30-10)/(50-10) = 0.5
3. c) 因为它是基于距离的算法，对特征尺度敏感
   KNN基于特征间距离计算
   不同尺度的特征在距离计算中权重不同
   通过缩放给所有特征赋予同等权重

PL20: 实践作业 (小组活动)



- 对实际数据集应用各种缩放技术
  - 提供的数据集: student_performance.csv
  - 应用不同缩放技术并比较结果
  - 讨论各方法的优缺点
- 实验对机器学习模型性能的影响
  - 分割为训练/测试集
  - 比较缩放前后模型性能
  - 测试至少3种模型 (KNN, 线性回归, SVM等)
  - 可视化结果并解释
- 提交内容:
  - Python代码 (.py文件)
  - 结果报告 (1-2页)
  - 下节课5分钟展示

PL21: 总结与下节课预告



今日学习内容

- 使用scikit-learn进行数据缩放的方法
- 实际实现各种缩放技术
  - Min-Max归一化
  - Z-score标准化
  - 对数变换
  - Robust缩放等
- 缩放对机器学习模型性能的影响
- 自定义缩放函数实现方法

下节课预告

- 数据编码实践
- 分类数据预处理方法
- LabelEncoder, OneHotEncoder, get_dummies使用方法

数据缩放相关术语 (Data Scaling Related Terms)



1. Data Scaling: 数据缩放Shùjù suōfàng): 데이터 스케일링
2. Normalization: 归一化(Guīyīhuà): 정규화
3. Standardization: 标准化(Biāozhǔnhuà): 표준화
4. Min-Max Normalization: Min-Max归一化(Min-Max guīyīhuà)
5. Z-score Standardization: Z-score标准化(Z-score biāozhǔnhuà):
6. Log Transformation: 对数变换(Duìshù biànhuàn): 로그 변환
7. Robust Scaling: 鲁棒缩放(Lǔbàng suōfàng): 로버스트 스케일링
8. MaxAbs Scaling: 绝对值最大缩放(Juéduìzhí zuìdà suōfàng
9. Quantile Transformation: 分位数变换(Fènwèishù biànhuàn): 분위수 변환

统计和数学术语 (Statistical and Mathematical Terms)



1. Mean: 均值(Jūnzhí): 평균
2. Standard Deviation: 标准差(Biāozhǔnchā): 표준편차
3. Median: 中位数(Zhōngwèishù): 중앙값
4. IQR (Interquartile Range): 四分位范围(Sìfènwèi fànwéi)
5. Outlier: 离群值(Líqún zhí): 이상치
6. Normal Distribution: 正态分布(Zhèngtài fēnbù): 정규분포
7. Skewed Distribution: 偏斜分布(Piānxié fēnbù): 편향 분포
8. Linear Transformation: 线性变换(Xiànxìng biànhuàn): 선형 변환
9. Non-linear Transformation: 非线性变换(Fēi xiànxìng biànhuàn)

机器学习相关术语(Machine Learning Related Terms)



1. Machine Learning: 机器学习(Jīqì xuéxí): 머신러닝
2. Feature: 特征(Tèzhēng): 특성
3. Gradient Descent: 梯度下降(Tīdù xiàjiàng): 경사 하강법
4. Regularization Term: 正则化项(Zhèngzé huà xiàng): 정규화항
5. KNN (K-Nearest Neighbors): K-最近邻(K-zuì jìn lín): K-최근접 이웃
6. Decision Tree: 决策树(Juécè shù): 의사결정나무
7. SVM (Support Vector Machine): 支持向量机(Zhīchí xiàngliàng jī)
8. Neural Network: 神经网络(Shénjīng wǎngluò): 신경망
9. Linear Regression: 线性回归(Xiànxìng huíguī): 선형 회귀
10. Logistic Regression: 逻辑回归(Luójí huíguī): 로지스틱 회귀
11. PCA (Principal Component Analysis): 主成分分析

数据处理相关术语(Data Processing Related Terms)



1. Data Preprocessing: 数据预处理(Shùjù yùchǔlǐ): 데이터 전처리
2. Training Set: 训练集(Xùnliàn jí): 훈련 세트
3. Test Set: 测试集(Cèshì jí): 테스트 세트
4. Data Leakage: 数据泄露(Shùjù xièlòu): 데이터 유출
5. Image Processing: 图像处理(Túxiàng chǔlǐ): 이미지 처리
6. Pixel Value: 像素值(Xiàngsù zhí): 픽셀값
7. Sparse Matrix: 稀疏矩阵(Xīshū jùzhèn): 희소 행렬
8. Categorical Variable: 分类变量(Fènlèi biànliàng): 범주형 변수
9. Encoding: 编码(Biānmǎ): 인코딩
10. Label Encoding: 标签编码(Biāoqiān biānmǎ): 레이블 인코딩
11. One-hot Encoding: 独热编码(Dúrè biānmǎ): 원-핫 인코딩