数据预处理

—— 归一化与标准化实践工作册 ——

Python 实战练习

姓名：_____　　学号：_____

日期：_____　　班级：_____

第一部分：理论回顾

一、什么是特征缩放?

在机器学习和数据分析中，不同特征的量纲和范围往往存在巨大差异。例如，年龄的取值范围可能是0-100，而收入的取值可能是0-1000000。特征缩放（Feature Scaling）是将数据转换到统一尺度的过程。

> ✨ **关键概念**
>
> 特征缩放的两种主要方法：归一化（Normalization / Min-Max Scaling）和标准化（Standardization / Z-Score Scaling）。它们的适用场景不同，选择时需要根据数据分布和算法需求来判断。

二、归一化（Min-Max Normalization）

归一化是将数据线性映射到 [0, 1] 区间的方法。公式如下：

```
X_norm = (X - X_min) / (X_max - X_min)
```

其中：X_min 和 X_max 分别是该特征的最小值和最大值。

特点：

- 结果范围固定在 [0, 1]
- 对异常值敏感（异常值会压缩正常数据的范围）
- 适用于均匀分布或需要固定范围的场景（如图像处理、神经网络）

三、标准化（Z-Score Standardization）

标准化是将数据转换为均值为0、标准差为1的分布。公式如下：

```
X_std = (X - μ) / σ
```

其中：μ 是均值，σ 是标准差。

特点：

- 结果无固定范围，但以 0 为中心
- 对异常值相对不敏感
- 适用于正态分布或近似正态分布的数据（如 SVM、逻辑回归、KNN）

四、对比摘要

对比维度	归一化 (Min-Max)	标准化 (Z-Score)
公式	(X-Xmin)/(Xmax-Xmin)	(X-μ)/σ
输出范围	[0, 1]	无固定范围
异常值影响	影响大	影响较小
数据分布假设	无	假设近似正态分布
典型算法	神经网络、KNN、图像处理	SVM、逻辑回归、PCA
sklearn类	MinMaxScaler	StandardScaler

理论练习

练习 1.1 填空题

1. 归一化将数据映射到 _____ 区间，而标准化将数据转换为均值为 _____ 、标准差为 _____ 的分布。

2. 归一化对 _____ 敏感，而标准化对异常值相对 _____ 。

3. 在 scikit-learn 中，归一化使用 _____ 类，标准化使用 _____ 类。

练习 1.2 手动计算

给定数据集：X = [10, 20, 30, 40, 50]

(a) 请手动计算归一化后的值：

计算过程：

(b) 请手动计算标准化后的值（提示：μ=30, σ≈14.14）：

计算过程：

练习 1.3 场景分析

判断以下场景应该使用归一化还是标准化，并简要说明理由：

(a) 使用KNN算法进行客户分类，特征包括年龄、收入、消费次数：

你的答案：

(b) 对图片像素值（0-255）进行处理，输入到卷积神经网络：

你的答案：

(c) 使用PCA进行降维，数据包含异常值：

你的答案：

第二部分：基础实践

练习 2.1：环境准备与数据加载

运行以下代码，导入必要的库并创建示例数据集：

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.datasets import load_iris
import matplotlib.pyplot as plt

# 加载 Iris 数据集
iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target

print(df.head())
print(df.describe())
```

任务：

1. 运行以上代码，观察数据集的基本结构和统计信息。

2. 记录各特征的最大值、最小值、均值和标准差：

特征名	最小值	最大值	均值	标准差
sepal length (cm)				
sepal width (cm)				
petal length (cm)				
petal width (cm)				

3. 不同特征的值域是否一致？这会带来什么问题？

你的观察：

练习 2.2：使用 MinMaxScaler 进行归一化

补全以下代码，完成归一化操作：

```
# 创建 MinMaxScaler 对象
scaler_mm = _____________  # TODO: 初始化MinMaxScaler

# 选择数值特征列
features = df[iris.feature_names]

# 拟合并转换数据
features_normalized = _____________  # TODO: 使用fit_transform

# 转换为DataFrame方便查看
df_norm = pd.DataFrame(features_normalized,
                       columns=iris.feature_names)

print('\n归一化后的数据：')
print(df_norm.describe())
```

任务：

1. 填写代码中的 TODO 部分并运行。

2. 验证归一化后各特征的最小值是否为 0，最大值是否为 1：

验证结果：

参考答案

scaler_mm = MinMaxScaler() 和 features_normalized = scaler_mm.fit_transform(features)

练习 2.3：使用 StandardScaler 进行标准化

补全以下代码，完成标准化操作：

```
# 创建 StandardScaler 对象
scaler_std = _____________  # TODO: 初始化StandardScaler

# 拟合并转换数据
features_standardized = _____________  # TODO: 使用fit_transform

# 转换为DataFrame
df_std = pd.DataFrame(features_standardized,
                      columns=iris.feature_names)

print('\n标准化后的数据：')
print(df_std.describe())
```

任务：

1. 填写代码中的 TODO 部分并运行。

2. 验证标准化后各特征的均值是否接近 0，标准差是否接近 1：

验证结果：

参考答案

scaler_std = StandardScaler() 和 features_standardized = scaler_std.fit_transform(features)

第三部分：可视化对比

练习 3.1：绘制分布直方图对比

运行以下代码，观察原始数据、归一化数据和标准化数据的分布差异：

```
fig, axes = plt.subplots(1, 3, figsize=(15, 4))

# 原始数据分布
axes[0].set_title('原始数据', fontsize=14)
for col in iris.feature_names:
    axes[0].hist(df[col], alpha=0.6, label=col, bins=20)
axes[0].legend(fontsize=8)

# 归一化后分布
axes[1].set_title('Min-Max 归一化', fontsize=14)
for col in iris.feature_names:
    axes[1].hist(df_norm[col], alpha=0.6, label=col, bins=20)
axes[1].legend(fontsize=8)

# 标准化后分布
axes[2].set_title('Z-Score 标准化', fontsize=14)
for col in iris.feature_names:
    axes[2].hist(df_std[col], alpha=0.6, label=col, bins=20)
axes[2].legend(fontsize=8)

plt.tight_layout()
plt.savefig('distribution_comparison.png', dpi=150)
plt.show()
```

观察与思考：

1. 原始数据中，不同特征的值域有什么不同？

你的观察：

2. 归一化和标准化后，数据分布的形状是否发生了变化？为什么？

你的分析：

练习 3.2：绘制箱线图对比

自己编写代码，绘制三组箱线图（原始数据 / 归一化 / 标准化），观察异常值的变化：

提示：使用 plt.subplots(1, 3) 和 DataFrame.boxplot()

在此写出你的代码：

观察结果，箱线图中的异常值在三种情况下有什么不同？

你的分析：

第四部分：进阶实践

练习 4.1：fit 与 transform 的区别

在实际项目中，fit() 和 transform() 的正确使用至关重要。请补全以下代码：

```
from sklearn.model_selection import train_test_split

# 划分训练集和测试集
X_train, X_test = train_test_split(
    features, test_size=0.3, random_state=42)

scaler = StandardScaler()

# TODO: 对训练集进行 fit_transform
X_train_scaled = _____________

# TODO: 对测试集只进行 transform（不能重新fit）
X_test_scaled = _____________

print('训练集均值:', X_train_scaled.mean(axis=0))
print('测试集均值:', X_test_scaled.mean(axis=0))
```

思考题：

1. 为什么测试集只能用 transform() 而不能用 fit_transform()?

你的答案：

2. 如果测试集也使用 fit_transform()，会导致什么问题?

你的答案：

参考答案

X_train_scaled = scaler.fit_transform(X_train) 和 X_test_scaled = scaler.transform(X_test)。测试集必须使用训练集的统计参数，否则会造成数据泄露（Data Leakage）。

练习 4.2：异常值对缩放的影响

运行以下代码，观察异常值如何影响缩放结果：

```
# 创建含异常值的数据
data_normal = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
data_outlier = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 1000])

# 归一化
mm = MinMaxScaler()
norm_normal = mm.fit_transform(
    data_normal.reshape(-1, 1)).flatten()
norm_outlier = mm.fit_transform(
    data_outlier.reshape(-1, 1)).flatten()

# 标准化
ss = StandardScaler()
std_normal = ss.fit_transform(
    data_normal.reshape(-1, 1)).flatten()
std_outlier = ss.fit_transform(
    data_outlier.reshape(-1, 1)).flatten()

print('无异常值 - 归一化:', np.round(norm_normal, 3))
print('有异常值 - 归一化:', np.round(norm_outlier, 3))
print('无异常值 - 标准化:', np.round(std_normal, 3))
print('有异常值 - 标准化:', np.round(std_outlier, 3))
```

分析任务：

1. 有异常值时，归一化后正常数据点的分布发生了什么变化？

你的观察：

2. 相比之下，标准化对异常值的表现如何？

你的分析：

3. 基于以上观察，如果数据中存在异常值，你会推荐哪种缩放方法？

你的建议：

练习 4.3：自定义缩放范围

MinMaxScaler 默认将数据缩放到 [0,1]，但也可以自定义范围。请完成以下任务：

(a) 将数据缩放到 [-1, 1] 范围：

```
# 提示：MinMaxScaler(feature_range=(min, max))
scaler_custom = _____________  # TODO
features_custom = scaler_custom.fit_transform(features)

df_custom = pd.DataFrame(features_custom,
                         columns=iris.feature_names)
print(df_custom.describe())
```

在此写出完整代码：

(b) 思考：在什么场景下你可能需要 [-1, 1] 而不是 [0, 1] 的范围？

你的思考：

第五部分：综合挑战

练习 5.1：完整的数据预处理流程

请独立完成以下完整的数据预处理流程：

数据集：假设有以下学生成绩数据：

姓名	数学	英语	物理	月生活费(元)
张三	85	72	90	1500
李四	92	88	78	2200
王五	76	95	85	1800
赵六	68	81	92	3500
孙七	95	69	71	1200

任务要求：

Step 1: 用 pandas 创建该数据的 DataFrame

Step 2: 分别对数据进行归一化和标准化

Step 3: 绘制每个特征的对比图（原始 / 归一化 / 标准化）

Step 4: 分析“月生活费”与其他特征在缩放前后的对比

在此写出完整代码：

分析与结论：

练习 5.2：实际数据集处理

加载 sklearn 自带的红酒质量数据集，完成完整的缩放流程：

```
from sklearn.datasets import load_wine

# 加载数据
wine = load_wine()
df_wine = pd.DataFrame(wine.data, columns=wine.feature_names)

# TODO: 请完成以下任务
# 1. 查看数据基本统计信息
# 2. 划分训练集和测试集 (test_size=0.2)
# 3. 分别用归一化和标准化处理训练集和测试集
#    （注意：测试集只能用transform）
# 4. 绘制对比图
# 5. 总结两种方法的差异
```

在此写出完整代码：

总结报告：

第六部分：自我评估

请根据你的学习情况，诚实填写以下自评表：

评估项目	完全掌握	基本掌握	需要复习
理解归一化的原理和公式	□	□	□
理解标准化的原理和公式	□	□	□
能够区分两种方法的适用场景	□	□	□
能够使用 MinMaxScaler 进行归一化	□	□	□
能够使用 StandardScaler 进行标准化	□	□	□
理解 fit / transform 的区别	□	□	□
能够绘制可视化对比图	□	□	□
理解异常值对缩放的影响	□	□	□

学习心得与疑问：

请写下你的学习心得、疑问或需要进一步探索的内容：