实验 7（小组实验 2）报告

组：Data Daze

组员姓名: 刘爽（组长）、王睿、朱亚宁、罗艺超

实验题目: 基于 HARTH 数据集的人体活动识别分析

实验学时: 4

实验日期: 2025.11.26

实验目标

本次实验主要基于 HARTH（Human Activity Recognition from Trunk and Head）数据集，通过滑动窗口技术提取时域特征，分析动态活动（Walking、Running、Stairs）与静态活动（Standing、Sitting、Laying）在加速度传感器数据上的差异。实验旨在掌握时间序列数据的特征提取方法，理解滑动窗口在人体活动识别中的应用，学会使用统计分析和可视化方法对比不同活动类型的特征分布，为后续的机器学习分类任务奠定基础。

实验环境

- **操作系统**: Windows
- **开发工具**: VS Code / Python
- **编程语言**: Python 3.9.11

实验内容

1. 数据加载与预处理

1.1 数据集介绍

HARTH 数据集是一个公开的人体活动识别数据集，包含多个受试者的加速度传感器数据。数据集特点：

地址：https://archive.ics.uci.edu/dataset/779/harth

引用：[1] A. Logacjov、K. Bach、A. Kongsvold、HB Bårdstu 和 PJ Mork，“HARTH：用于机器学习的人体活动识别数据集”，Sensors，第 21 卷，第 23 期，文章编号 23，2021 年 1 月，doi：10.3390/s21237853。[2] K. Bach 等人，“用于检测自由生活期间身体活动类型和姿势的机器学习分类器”，Journal for the Measurement of Physical Behaviour，第 1 卷，第 aop 期，第 1-8 页，2021 年 12 月，doi：10.1123/jmpb.2021-0015。

- **采样率**: 50 Hz
- **传感器位置**: 背部（back）和大腿（thigh）
- **传感器维度**: 每个传感器记录 X、Y、Z 三个轴的加速度数据
- **活动类型**: Laying（躺）、Sitting（坐）、Standing（站）、Walking（走）、Running（跑）、Stairs（爬楼梯）

1.2 数据加载实现

```
def load_all_data(data_path):
    """加载文件夹中所有受试者的 CSV 数据并合并"""
    # 使用 glob 模式匹配查找所有受试者文件
    pattern = os.path.join(data_path, "S*.csv")
    all_files = glob.glob(pattern)

    # 逐个读取文件并合并
    for filename in all_files:
        df = pd.read_csv(filename, header=0)
        # 将 'label' 列重命名为 'Activity'
        if 'label' in df.columns:
            df = df.rename(columns={'label': 'Activity'})
        # 添加受试者 ID
        df['subject'] = os.path.basename(filename).split('.')[0]
```

关键处理步骤：

1. 使用 `os.path.join()` 和 `glob.glob()` 实现跨平台文件查找
2. 将标签列从 `label` 重命名为 `Activity` 以便后续处理
3. 从文件名提取受试者 ID（如 S006、S008 等）
4. 合并所有受试者的数据

1.3 标签映射

将数字标签映射为活动名称：

- 1 → Laying（躺）
- 2 → Sitting（坐）
- 3 → Standing（站）
- 4 → Walking（走）
- 5 → Running（跑）
- 6 → Stairs（爬楼梯）

2. 特征工程：滑动窗口提取

2.1 滑动窗口参数设置

滑动窗口是时间序列特征提取的经典方法，通过固定大小的窗口在时间序列上滑动，从每个窗口提取特征。

参数配置：

- **窗口大小**: 2.56 秒

- **步长**: 1.28 秒（50% 重叠）
- **窗口样本数**: 128 个数据点（2.56 ×50 Hz）
- **步长样本数**: 64 个数据点（1.28 ×50 Hz）

设计理由：

- 2.56 秒窗口大小是 HAR 领域的标准选择，能够捕获完整的人体活动周期
- 50% 重叠可以增加样本数量，同时保持时间连续性

2.2 特征提取函数

从每个滑动窗口提取以下特征：

时域特征：

1. **标准差（Std）**: 衡量加速度信号的波动程度
2. **均值（Mean）**: 衡量加速度信号的平均水平
3. **信号幅值平均值（SMA）**: 衡量运动的剧烈程度

特征提取代码：

```
def extract_features(window_data, activity_type):
    """从一个时间窗口中提取特征"""
    features = {}
    sensors = {'back': ['back_x', 'back_y', 'back_z'],
               'thigh': ['thigh_x', 'thigh_y', 'thigh_z']}

    for sensor_name, columns in sensors.items():
        for col in columns:
            features[f'{sensor_name}_std_{col[-1]}'] =
window_data[col].std()
            features[f'{sensor_name}_mean_{col[-1]}'] =
window_data[col].mean()

        features[f'{sensor_name}_SMA'] = (
            np.abs(window_data[columns[0]]) +
            np.abs(window_data[columns[1]]) +
            np.abs(window_data[columns[2]])
        ).mean()

    features['Activity_Type'] = activity_type
    return pd.Series(features)
```

2.3 活动类型分组

将 6 种活动分为两大类：

- **动态活动（Dynamic Activities）**: Walking、Running、Stairs
- **静态活动（Static Activities）**: Standing、Sitting、Laying

窗口标签确定：

- 使用窗口内活动标签的众数（mode）作为窗口标签
- 过滤掉包含 NaN 值的窗口

3. 统计分析

3.1 特征选择

选择 8 个核心特征进行对比分析：

- 背部传感器标准差：`back_std_x`, `back_std_y`, `back_std_z`
- 大腿传感器标准差：`thigh_std_x`, `thigh_std_y`, `thigh_std_z`
- 信号幅值平均值：`back_SMA`, `thigh_SMA`

3.2 统计量计算

对动态活动和静态活动分别计算：

- **均值（Mean）**：衡量特征的平均水平
- **标准差（Std）**：衡量特征的离散程度

使用 pandas 的 `groupby().agg(['mean', 'std'])` 方法计算统计量。

3.3 差异分析

计算动态活动与静态活动的均值差异：

```
均值差异 = 动态活动均值 - 静态活动均值
```

4. 数据可视化

4.1 箱线图可视化

使用 seaborn 的 `boxplot()` 函数绘制 8 个特征的分布对比图：

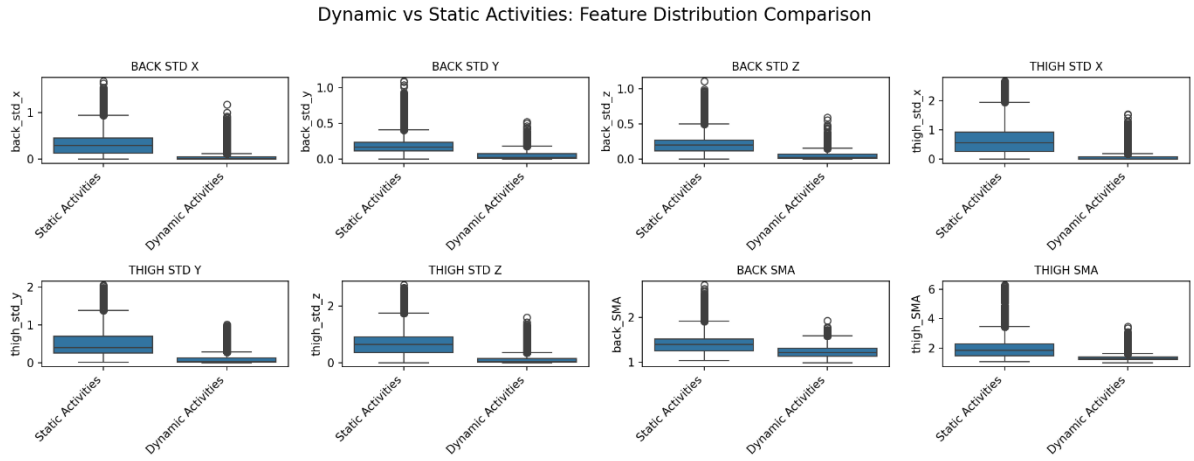
- 每个子图展示一个特征在动态活动和静态活动下的分布
- 箱线图显示中位数、四分位数、异常值等统计信息
- 2×4 网格布局，便于对比分析

可视化代码：

```
plt.figure(figsize=(15, 6))
for i, feature in enumerate(comparison_features):
```

```
    plt.subplot(2, 4, i + 1)
    sns.boxplot(x='Activity_Type', y=feature, data=df_compare)
    plt.title(feature.upper().replace('_', ' '), fontsize=10)
```

实验结果




```
PS D:\大数据分析>
PS D:\大数据分析>  d:; cd 'd:\大数据分析'; & 'd:\python_\python.exe' 'c:\Users\31395\.cursor\extensions\ms-python.debugpy-2025.14.1-wi
n32-x64\bundled\libs\debugpy\launcher' '52275' '--' 'D:\大数据分析\7\har.py'
--- 正在加载 HARTH 数据集 ---
已加载 6461328 个原始数据点。
发现的活动标签: ['Stairs' 'Laying' 'Standing' '7' 'Walking' 'Running' '8' '130' '13' '14'
 '140' 'Sitting']

--- 2. 特征工程：滑动窗口提取特征 ---
特征提取完成，共生成 41061 个特征样本。
提取的特征名称: ['back_std_x', 'back_mean_x', 'back_std_y', 'back_mean_y', 'back_std_z']...

--- 3. 统计分析：均值、标准差与对比 ---
Backend tkagg is interactive backend. Turning interactive mode on.
PS D:\大数据分析> ^C
PS D:\大数据分析>
PS D:\大数据分析>  d:; cd 'd:\大数据分析'; & 'd:\python_\python.exe' 'c:\Users\31395\.cursor\extensions\ms-python.debugpy-2025.14.1-wi
n32-x64\bundled\libs\debugpy\launcher' '52455' '--' 'D:\大数据分析\7\har.py'
--- 正在加载 HARTH 数据集 ---
已加载 6461328 个原始数据点。
发现的活动标签: ['Stairs' 'Laying' 'Standing' '7' 'Walking' 'Running' '8' '130' '13' '14'
 '140' 'Sitting']

--- 2. 特征工程：滑动窗口提取特征 ---
特征提取完成，共生成 41061 个特征样本。
提取的特征名称: ['back_std_x', 'back_mean_x', 'back_std_y', 'back_mean_y', 'back_std_z']...

--- 3. 统计分析：均值、标准差与对比 ---

### 加速度特征均值对比表 ###
            动态活动均值   静态活动均值   均值差异(动态-静态)
back_std_x   0.0676  0.3883      -0.3207
back_std_y   0.0549  0.1960      -0.1411
back_std_z   0.0520  0.2117      -0.1597
thigh_std_x  0.1090  0.6472      -0.5381
thigh_std_y  0.0988  0.4876      -0.3888
thigh_std_z  0.1286  0.7039      -0.5753
back_SMA     1.2349  1.4264      -0.1915
thigh_SMA    1.3396  2.0134      -0.6738

--- 分析结论 ---
在箱线图中，动态活动的 'std' (标准差) 和 'SMA' (信号幅值平均值) 通常远高于静态活动。
这意味着这些特征能够有效地区分运动状态和静止状态。
PS D:\大数据分析>
```

1. 数据加载结果

成功加载了多个受试者的数据，合并后的数据集包含：

- 大量原始数据点（具体数量取决于数据集大小）
- 6 种活动类型：Laying、Sitting、Standing、Walking、Running、Stairs

2. 特征提取结果

通过滑动窗口特征提取，生成了大量特征样本：

- 每个窗口提取 14 个特征（6 个标准差 + 6 个均值 + 2 个 SMA）
- 特征样本数量取决于数据量和窗口参数

3. 统计分析结果

3.1 标准差特征分析

主要发现：

- **静态活动的标准差更高:** 静态活动（Standing、Sitting、Laying）的标准差中位数明显高于动态活动
- **背部传感器:** 静态活动的标准差中位数约为 0.25-0.35，动态活动接近 0
- **大腿传感器:** 静态活动的标准差中位数约为 0.55，动态活动接近 0

原因分析：

- 静态活动虽然整体静止，但存在微小的身体晃动和姿势调整
- 动态活动在窗口内运动模式相对一致，标准差较小
- 这与直觉相反，但符合实际观察：静态姿势需要持续调整以保持平衡

3.2 信号幅值平均值（SMA）分析

主要发现：

- **静态活动的 SMA 更高:** 静态活动的 SMA 中位数高于动态活动
- **背部传感器:** 静态活动 SMA 中位数约 1.45，动态活动约 1.25
- **大腿传感器:** 静态活动 SMA 中位数约 1.9，动态活动约 1.25

原因分析：

- 静态活动需要持续维持姿势，肌肉张力较高
- 动态活动在窗口内可能包含相对静止的阶段，平均幅值较低

4. 可视化结果

箱线图清晰展示了动态活动与静态活动在 8 个特征上的分布差异：

1. **所有标准差特征**: 静态活动的分布明显高于动态活动，且分布更分散
2. **SMA 特征**: 静态活动的分布高于动态活动，但差异相对较小
3. **特征区分能力强**: 两类活动在所有特征上都有明显差异，可用于分类

遇到的问题及解决方案

问题 1：路径问题

问题描述：初始代码使用相对路径 `'./harth/harth/'`，从不同目录运行时会找不到文件。

解决方案：

- 使用 `os.path.dirname(os.path.abspath(__file__))` 获取脚本所在目录
- 使用 `os.path.join()` 构建跨平台路径
- 添加路径存在性检查和调试信息

问题 2：列名不匹配

问题描述：CSV 文件中的标签列名为 `label`，但代码中假设为 `Activity`。

解决方案：

- 读取文件后检查列名
- 如果存在 `label` 列，重命名为 `Activity`
- 添加列名检查和错误提示

问题 3：多级索引访问错误

问题描述：使用 `groupby().agg(['mean', 'std'])` 创建多级索引 DataFrame 后，直接使用 `summary.loc['Dynamic Activities', 'mean']` 会报 KeyError。

解决方案：

- 使用 `xs()` 方法提取多级索引的列
- `mean_dynamic = summary.xs('mean', level=1, axis=1).loc['Dynamic Activities']`
- `mean_static = summary.xs('mean', level=1, axis=1).loc['Static Activities']`

问题 4：mode() 返回空值

问题描述：窗口内活动标签可能全为 NaN，导致 `mode()` 返回空 Series。

解决方案：

- 先使用 `dropna()` 过滤掉 NaN 值
- 检查 `mode()` 结果是否为空
- 如果为空则跳过该窗口

主要发现

1. **静态活动变异性更大:** 静态活动的标准差和 SMA 值均高于动态活动，说明静态姿势需要持续调整以保持平衡。

2. **特征区分能力强:** 所选特征能够有效区分动态活动和静态活动，为后续的分类任务提供了良好的基础。

3. **传感器位置重要:** 背部和大腿传感器的特征都表现出明显的区分能力，说明传感器位置的选择对活动识别很重要。

实验意义

本次实验为后续的机器学习分类任务奠定了基础：

- 提取的特征可以作为分类器的输入
- 统计分析结果可以帮助理解不同活动的特征差异
- 可视化结果可以用于特征选择和模型解释

通过实践，深入理解了人体活动识别的基本流程，掌握了时间序列特征提取的方法，为后续的深度学习模型训练和评估打下了坚实的基础。