

实验 4 基于线性回归算法的儿童身高预测实习指导书

一、实验基本信息

项目	内容
实验名称	基于线性回归算法的儿童身高预测
授课课程	大数据分析 / 机器学习基础
授课班级	(按实际班级填写)
实验学时	2 学时 (90 分钟)
实验类型	操作性 + 应用性实验
实验目标	1. 掌握线性回归模型的数据预处理操作 (含数据清洗、特征理解、训练集逻辑处理) ; 2. 掌握 scikit-learn 中线性回归模型的构建、训练与预测流程; 3. 理解模型评估与结果可视化的核心逻辑, 培养数据驱动预测的科研思维与细致操作习惯。

二、实验原理

1. 线性回归核心原理

线性回归假设自变量 (如年龄、父母身高) 与因变量 (儿童身高) 存在线性关系, 通过最小二乘法求解最优参数 (回归系数), 使得预测值与真实值的“残差平方和”最小。模型形式为:

$y = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$

其中, $x_1, x_2, \dots, x_n$ 是自变量 (如年龄、性别、亲属身高), $w_1, w_2, \dots, w_n$ 是回归系数, b 是截距, y 是预测的儿童身高。

2. 数据预处理逻辑

为简化问题, 实验假设“儿童 18 岁后身高不再增长”, 因此需对**超过 18 岁的样本**进行预处理 (将年龄修正为 18 岁), 保证模型符合业务逻辑; 同时, 需明确特征含义 (如性别用“0 = 女, 1 = 男”编码), 确保输入模型的特征与训练数据一致。

3. 模型构建与预测流程

1. 用 `scikit-learn` 的 `LinearRegression` 创建模型;
2. 通过 `fit(X, y)` 方法, 利用训练数据拟合模型 (求解最优参数);
3. 对新样本, 用 `predict(X_new)` 输出预测身高;
4. 需注意新样本的**特征维度、含义必须与训练数据严格一致**, 否则预测无意义。

三、实验环境准备

1. 硬件要求

计算机内存 $\geq 4\text{GB}$, 硬盘剩余空间 $\geq 10\text{GB}$ (用于存储实验代码、数据及临时文件)。

2. 软件要求

1. 操作系统: Windows 10/11 (64 位)、macOS 10.15+ 或 Linux (Ubuntu 20.04+);
2. 软件套件: Anaconda3 (2024.02 及以上版本, 预装 `numpy`、`scikit-learn`);
3. 开发工具: Spyder (Anaconda 自带, 确保 IPython 控制台可正常运行代码)。

1. 预检查步骤 (避免实验报错)

1. 打开 Spyder/VS Code 的终端, 输入以下代码, 验证库是否正常导入:

```
import numpy as np
from sklearn import linear_model
print("NumPy 版本: ", np.__version__)      # 需 $\geq 1.20$ 
print("scikit-learn 版本: ", linear_model.__version__) # 需 $\geq 1.0$ 
```

1. 若提示 `ModuleNotFoundError` (如缺少 `scikit-learn`): 打开 Anaconda Prompt, 执行 `conda install scikit-learn` 安装;

2. 若版本过低：执行 `conda update numpy scikit-learn` 升级。

四、实验内容与操作步骤

本实验分 3 个核心模块，每个模块遵循“任务描述→操作步骤→细节验证→错误排查”流程，通过强制验证与逻辑检查，培养“严谨建模、细致验证”的科研习惯。

模块 1：数据预处理与理解

操作目标

明确儿童身高预测的特征含义，完成“18 岁后年龄修正”的预处理逻辑，确保输入模型的数据符合业务假设。

操作步骤

1. 特征与数据含义分析（培养特征敏感性）

1. 在 Spyder 编辑器中新建文件，命名为 `工工 2241_学号_姓名_LinearRegExp4.py`（如 `工工 2241_220101_张三_LinearRegExp4.py`，禁止含中文空格）；
2. 输入以下代码（含注释），理解训练数据的特征与标签：

```
import copy
import numpy as np
from sklearn import linear_model

# 任务 1-1：理解训练数据特征与标签
# 训练数据 x：每一行是 1 个样本，特征依次为：
# 儿童年龄，性别（0=女，1=男），父亲身高，母亲身高，祖父身高，祖母身高，外祖父身高，外祖母身高
x = np.array([
    [1, 0, 180, 165, 175, 165, 170, 165],
    [3, 0, 180, 165, 175, 165, 173, 165],
    [4, 0, 180, 165, 175, 165, 170, 165],
    [6, 0, 180, 165, 175, 165, 170, 165],
    [8, 1, 180, 165, 175, 167, 170, 165],
```

```

    [10, 0, 180, 166, 175, 165, 170, 165],
    [11, 0, 180, 165, 175, 165, 170, 165],
    [12, 0, 180, 165, 175, 165, 170, 165],
    [13, 1, 180, 165, 175, 165, 170, 165],
    [14, 0, 180, 165, 175, 165, 170, 165],
    [17, 0, 170, 165, 175, 165, 170, 165]
])
# 训练数据 y: 儿童身高 (单位: cm)
y = np.array([60, 90, 100, 110, 130, 140, 150, 164, 160, 163, 168])
# 细节验证: 检查数据形状与特征含义
print("=== 训练数据验证 ===")
print("x 的形状 (样本数×特征数): ", x.shape) # 预期: (11, 8)
print("y 的形状 (标签数): ", y.shape)       # 预期: (11,)
print("第 1 个样本特征: 年龄={}, 性别={}, 父亲身高={}cm".format(
    x[0, 0], x[0, 1], x[0, 2]
)) # 预期: 年龄=1, 性别=0, 父亲身高=180cm
print("第 1 个样本真实身高: cm".format(y[0])) # 预期: 60cm

```

1. 运行代码, 若 `x.shape` 不为 `(11, 8)`, 检查 `x` 的数组是否漏写 / 多写行; 若特征含义对应错误, 重新核对注释与数组列顺序。
2. **18 岁后年龄修正逻辑 (培养业务 - 数据对齐习惯)**
 1. 模拟 “年龄 > 18 岁” 的样本, 测试预处理逻辑:

```

# 任务 1-2: 18 岁后年龄修正预处理
# 模拟 3 个待预测样本 (xs), 特征含义与 x 一致
xs = np.array([
    [10, 0, 180, 165, 175, 165, 170, 165], # 年龄 10 (<18, 无需修正)
    [17, 1, 173, 153, 175, 161, 170, 161], # 年龄 17 (<18, 无需修正)
    [34, 0, 170, 165, 170, 165, 170, 165]  # 年龄 34 (>18, 需修正为 18)
])

```

```

# 预处理：遍历每个样本，修正年龄>18 的情况
xs_processed = []
for item in xs:
    item_copy = copy.deepcopy(item) # 深拷贝，避免修改原数据
    if item_copy[0] > 18:
        item_copy[0] = 18 # 年龄>18 → 修正为 18
    xs_processed.append(item_copy)
xs_processed = np.array(xs_processed)
# 细节验证：检查预处理结果
print("\n=== 预处理结果验证 ===")
print("原始待预测样本：")
print(xs)
print("\n预处理后样本：")
print(xs_processed)
# 重点检查第 3 个样本（原年龄 34）
print("\n 第 3 个样本预处理后年龄：", xs_processed[2, 0]) # 预期：18

```

1. 运行代码，若第 3 个样本年龄未修正为 18，检查 `if item_copy[0] > 18` 的逻辑是否写反（如误写为 `<`）；若深拷贝失效，检查 `copy.deepcopy` 是否正确导入（`import copy` 是否存在）。

模块 2：线性回归模型构建与训练

操作目标

掌握 `scikit-learn` 中 `LinearRegression` 的创建、训练流程，确保模型能拟合训练数据，为后续预测提供基础。

操作步骤

1. 模型创建与训练（培养建模流程意识）
 1. 继续添加代码，创建并训练线性回归模型：

```
# 任务 2-1：创建并训练线性回归模型
```

```

# 1. 初始化线性回归模型
lr = linear_model.LinearRegression()

# 2. 用训练数据拟合模型（求解最优参数）
lr.fit(x, y)

# 细节验证：检查模型参数（非必须，但可辅助理解）
print("\n=== 模型训练验证 ===")

print("模型系数（w1~w8）：", lr.coef_)      # 8 个特征对应 8 个系数
print("模型截距（b）：", lr.intercept_)    # 截距项

# 验证：用第 1 个训练样本预测，应接近真实值 y[0]
pred_train_0 = lr.predict(x[0].reshape(1, -1))[0]
print("第 1 个训练样本预测身高：{:.2f}cm，真实身高：cm".format(
    pred_train_0, y[0]
)) # 预期：预测值接近 60cm（允许小误差）

```

1. 运行代码，若 `lr.coef_` 长度不为 8，检查 `x` 的特征数是否为 8（`x.shape[1]` 是否为 8）；若第 1 个样本预测值与真实值偏差极大（如差几十 cm），检查 `x` 与 `y` 的对应关系是否错误（如样本行是否对齐）。

模块 3：儿童身高预测与结果分析

操作目标

用训练好的模型预测新样本身高，验证“18 岁后身高不再增长”的业务假设，培养“预测 - 验证 - 分析”的科研逻辑。

操作步骤

1. 新样本预测与结果验证（培养预测严谨性）
 1. 继续添加代码，用预处理后的 `xs_processed` 预测身高：

```

# 任务 3-1：新样本预测与结果分析
# 用训练好的模型预测预处理后的待预测样本
y_pred = lr.predict(xs_processed)
# 细节验证：打印预测结果并分析

```

```

print("\n=== 预测结果与分析 ===")
for i in range(len(xs)):
    age_original = xs[i, 0]
    age_used = xs_processed[i, 0]
    height_pred = y_pred[i]
    print("原始年龄: {}岁, 模型使用年龄: {}岁, 预测身高: {:.2f}cm".format(
        age_original, age_used, height_pred
    ))
# 重点分析: 年龄 34 的样本, 预测身高是否与年龄 18 的样本逻辑一致

```

1. 运行代码, 观察输出:

1. 年龄 10、17 的样本: 预测身高应随年龄增长而合理上升;
2. 年龄 34 的样本: 因被修正为 18 岁, 预测身高应与“年龄 18 且其他特征相似”的样本逻辑一致 (体现“18 岁后不再长高”的假设)。

若年龄 34 的预测身高远高于 / 低于预期, 检查预处理是否生效 (`xs_processed[2, 0]` 是否为 18) 或模型训练是否异常。

1. 业务假设验证 (培养数据 - 业务关联思维)

1. 手动构造“年龄 20, 其他特征与年龄 18 样本一致”的新样本, 测试预测结果:

```

# 任务 3-2: 验证“18 岁后身高不变”假设
# 构造样本: 年龄 20, 性别 0, 亲属身高与训练集第 10 个样本 (年龄 14) 一致
test_item = np.array([20, 0, 180, 165, 175, 165, 170, 165])
# 预处理: 修正年龄>18
test_item_processed = copy.deepcopy(test_item)
if test_item_processed[0] > 18:
    test_item_processed[0] = 18
# 预测
test_pred = lr.predict(test_item_processed.reshape(1, -1))[0]
print("\n=== 业务假设验证 ===")
print("构造样本 (年龄 20→18) 预测身高: {:.2f}cm".format(test_pred))

```

对比：找到训练集中年龄 18 左右的样本预测值，验证是否接近

1. 运行代码，若构造样本的预测身高与“年龄 18、特征相似”的样本预测值差异大，需回溯模型训练过程（如训练数据是否充分、特征是否合理）。

五、实验注意事项

1. 特征一致性

待预测样本的**特征顺序、含义必须与训练数据严格一致**（如第 1 列是年龄、第 2 列是性别...），否则模型会将“性别”误判为“年龄”，导致预测完全错误。

2. 深拷贝与数据污染

预处理时必须用 `copy.deepcopy` 复制样本（而非浅拷贝），否则修改 `item_copy` 会污染原数组 `xs`，导致后续逻辑混乱。

3. 年龄修正的业务逻辑

实验假设“18 岁后身高不再增长”是**简化后的业务规则**，需明确该规则的局限性（实际身高还受健康、环境等更多因素影响），但实验中必须严格执行该逻辑以保证模型一致性。

4. 模型过拟合 / 欠拟合

若训练样本预测准确但新样本偏差大，可能是模型过拟合（训练数据量太少）；若所有预测都偏差大，可能是特征与标签无线性关系或模型未正确训练，需检查数据或模型参数。

六、实验报告要求

1. 报告结构

需包含“实验目的”“实验原理（简述线性回归、数据预处理、模型训练预测逻辑）”“实验步骤与结果（附关键代码、输出截图）”“问题与解决方法”“实验总结”5 个部分，严格遵循教案报告规范。

2. 关键截图要求

1. 必附截图：训练数据 `x` 与 `y` 的验证输出、预处理前后的 `xs` 对比、模型系数与截距、新样本预测结果（含年龄修正前后的分析）；
2. 截图需标注清晰（如“图 1 训练数据验证”“图 3 预测结果分析”），包含代码片段与对应输出，证明为本人操作，杜绝截图造假。

1. “问题与解决方法”要求

需记录至少 2 个实验中遇到的真实问题及解决过程，示例：

1. “问题 1：预测结果全为同一数值；解决：检查发现 `x` 的特征维度与模型期望不一致（`x` 是 1 维数组，需用 `reshape(1, -1)` 转为 2 维），修改后预测正常”；

2. “问题 2：年龄修正后预测身高仍随年龄增长；解决：检查 `if item_copy[0] > 18` 逻辑，发现误写为 `>=`，导致 18 岁样本也被修正，改为 `>` 后符合预期”。
1. “实验总结”要求
1. 技术总结：提炼线性回归预测的核心流程（“数据预处理→模型训练→预测验证”）、`scikit-learn` 中 `LinearRegression` 的关键方法（`fit()`、`predict()`）；
2. 品质反思：反思实验中因“特征顺序错误”“深拷贝遗漏”导致的预测错误，提出改进措施（如预处理后用 `print` 逐行验证数据、建模前画特征与标签的散点图确认线性趋势）。

七、思考题（深化理解，关联实际应用）

1. 若想让模型同时考虑“18 岁前身高增长速率”和“18 岁后身高稳定”，需对模型或数据做哪些改进？（提示：特征工程，如添加“是否成年”标识特征）。
2. 实验中若训练数据包含“年龄 20、身高 175cm”的样本，模型还会强制将年龄 20 修正为 18 吗？这种情况下“18 岁后身高不变”的假设还成立吗？（提示：模型会学习训练数据中的规律，若训练数据有超 18 岁且身高增长的样本，假设会被打破）。
3. 如何用 `scikit-learn` 的 `mean_squared_error` 或 `r2_score` 评估模型在训练集上的拟合效果？请写出核心代码（提示：`from sklearn.metrics import mean_squared_error; mse = mean_squared_error(y_true, y_pred)`）。
4. 现实中儿童身高还受“营养摄入、运动习惯”等因素影响，若要让模型更准确，需如何扩展特征与数据？（提示：收集更多维度的特征数据，如每日蛋白质摄入量、每周运动时长等）。

八、实验考核标准

考核维度	考核要点（总分 100 分）	分值
环境准备（10 分）	<code>numpy</code> 与 <code>scikit-learn</code> 验证正确（版本符合要求，无库缺失错误）（5 分）；代码文件命名规范（5 分）	10

代码完成 (40 分)	模块 1-3 代码完整且运行正确 (每模块 13 分, 语法错误扣 3 分 / 处, 逻辑错误扣 5 分 / 处; 模块 3 加 1 分整体逻辑分)	40
结果验证 (20 分)	关键步骤有 <code>print()</code> 验证 (如数据形状、预处理结果、预测值), 结果与预期一致 (每处 5 分)	20
报告质量 (20 分)	结构完整 (5 分); 截图清晰标注 (5 分); “问题与解决方法” 真实详细 (10 分)	20
科研品质 (10 分)	代码注释规范 (3 分); 错误排查记录完整 (4 分); 实验总结体现逻辑反思 (3 分)	10