# 实验一：scikit-learn 入门与线性回归（2 学时）

## 学习目标

* 通过线性回归（LinearRegression）完成一个从数据到结果的完整流程。
* 会按步骤运行带有详细中文注释的脚本，理解每一步“在做什么、为什么这样做、得到什么结果”。
* 会阅读与解释三类常用指标：MAE、MSE、R²，并保存结果与图表。 （本章课时：**2 学时**）

## 先修知识与软件环境

* 系统：Windows 10 / cmd
* Python：3.12（已配置清华源）
* 依赖：numpy、pandas、scikit-learn、matplotlib、seaborn

## 实验准备

1. 创建并激活虚拟环境（cmd）：

python -m venv .venv  
\.venv\Scripts\activate.bat

1. 配置 pip 清华源（全局一次）：

pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple  
pip config set global.trusted-host pypi.tuna.tsinghua.edu.cn

1. 安装依赖：

pip install numpy pandas scikit-learn matplotlib seaborn

1. 创建输出目录：

mkdir outputs

## 核心知识要点

* 线性回归：用一条“最佳拟合直线/超平面”来预测连续数值。
* 数据切分：用一部分数据训练模型，另一部分数据测试模型的泛化能力。
* 指标解读：
  + MAE：平均“偏差”的绝对值，越小越好，直观易懂；
  + MSE：更惩罚“大误差”，越小越好；
  + R²：拟合程度（0~1），越接近 1 越好。

## 实验任务清单

1. 加载糖尿病数据集（内置，离线可用）。
2. 切分训练/测试集（8:2）。
3. 训练一个线性回归模型。
4. 评估并输出 MAE/MSE/R²，绘制“真实值 vs 预测值”散点图与残差分布图。
5. 保存模型与图表，便于写实验报告。

## 操作步骤

执行策略：只需运行给定命令/代码即可完成。代码已包含详细中文注释。

1. 新建脚本 exp1\_linear\_regression.py，将以下代码完整复制粘贴保存：

"""  
脚本名称：exp1\_linear\_regression.py  
用途：通过线性回归，演示从数据加载→切分→训练→评估→可视化→保存的完整流程。  
如何运行（Windows 10 + cmd）：  
 1) 激活虚拟环境： .\.venv\Scripts\activate.bat  
 2) 执行命令： python exp1\_linear\_regression.py --out-dir outputs --test-size 0.2 --seed 42  
  
输出：  
 - 指标文件：outputs\metrics.csv  
 - 散点图： outputs\pred\_scatter.png  
 - 残差图： outputs\residuals.png  
 - 模型文件：outputs\linear\_model.joblib  
"""  
  
import argparse  
import os  
from typing import Tuple  
  
import numpy as np  
import pandas as pd  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
from sklearn.datasets import load\_diabetes  
from sklearn.model\_selection import train\_test\_split  
from sklearn.linear\_model import LinearRegression  
from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score  
import joblib  
  
  
def load\_dataset() -> Tuple[np.ndarray, np.ndarray, list[str]]:  
 """加载内置的糖尿病数据集。  
 返回：特征 X、目标 y、特征名列表。  
 说明：该数据集体量小、离线可用，适合课堂演示。  
 """  
 ds = load\_diabetes()  
 X = ds.data # 特征矩阵（样本数 × 特征数）  
 y = ds.target # 回归目标（连续数值）  
 feature\_names = list(ds.feature\_names)  
 return X, y, feature\_names  
  
  
def evaluate(y\_true: np.ndarray, y\_pred: np.ndarray) -> tuple[float, float, float]:  
 """计算三种常用指标：MAE、MSE、R²。"""  
 mae = mean\_absolute\_error(y\_true, y\_pred)  
 mse = mean\_squared\_error(y\_true, y\_pred)  
 r2 = r2\_score(y\_true, y\_pred)  
 return mae, mse, r2  
  
  
def main():  
 # 解析命令行参数，便于在不改代码的情况下调整数据切分比例与随机种子。  
 parser = argparse.ArgumentParser(  
 description=(  
 "Exp1: Simple Linear Regression with "  
 "scikit-learn"  
 )  
 )  
 parser.add\_argument(  
 "--test-size",  
 type=float,  
 default=0.2,  
 help=(  
 "测试集比例，默认 0.2 即 8:2 切分"  
 ),  
 )  
 parser.add\_argument(  
 "--seed",  
 type=int,  
 default=42,  
 help=(  
 "随机种子，保证结果可复现"  
 ),  
 )  
 parser.add\_argument(  
 "--out-dir",  
 default="outputs",  
 help=(  
 "输出目录，用于保存图像与模型"  
 ),  
 )  
 args = parser.parse\_args()  
  
 # 1) 加载数据  
 print("[1/6] 加载数据集（糖尿病，内置，无需联网）...")  
 X, y, feature\_names = load\_dataset()  
 print(f" 样本数: {X.shape[0]}, 特征数: {X.shape[1]}")  
  
 # 2) 切分训练/测试集  
 print(  
 "[2/6] 按 8:2 切分训练集/测试集（可通过 --test-size 调整）..."  
 )  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(  
 X, y, test\_size=args.test\_size, random\_state=args.seed  
 )  
 print(f" 训练集: {X\_train.shape[0]} 条, 测试集: {X\_test.shape[0]} 条")  
  
 # 3) 训练线性回归模型  
 print("[3/6] 训练线性回归模型（无需调参）...")  
 model = LinearRegression()  
 model.fit(X\_train, y\_train)  
 print(" 训练完成。")  
  
 # 4) 在测试集上进行预测并评估  
 print("[4/6] 在测试集上预测并计算 MAE/MSE/R² 指标...")  
 y\_pred = model.predict(X\_test)  
 mae, mse, r2 = evaluate(y\_test, y\_pred)  
 print(f" MAE = {mae:.3f} MSE = {mse:.3f} R² = {r2:.3f}")  
  
 # 5) 可视化（真实值 vs 预测值；残差分布）并保存到输出目录  
 print("[5/6] 绘制并保存图像（散点图与残差分布）...")  
 os.makedirs(args.out\_dir, exist\_ok=True)  
 sns.set(style="whitegrid", font="sans-serif")  
  
 # 5.1 真实值 vs 预测值  
 plt.figure(figsize=(6, 6))  
 sns.scatterplot(x=y\_test, y=y\_pred, s=24, alpha=0.7)  
 lims = [min(y\_test.min(), y\_pred.min()), max(y\_test.max(), y\_pred.max())]  
 plt.plot(lims, lims, "r--", linewidth=1) # y=x 参考线  
 plt.xlabel("真实值 True")  
 plt.ylabel("预测值 Predicted")  
 plt.title("True vs Predicted (LinearRegression)")  
 plt.tight\_layout()  
 scatter\_path = os.path.join(args.out\_dir, "pred\_scatter.png")  
 plt.savefig(scatter\_path, dpi=150)  
 plt.close()  
  
 # 5.2 残差分布（真实-预测）  
 residuals = y\_test - y\_pred  
 plt.figure(figsize=(6, 4))  
 sns.histplot(residuals, kde=True, bins=30, color="#4C72B0")  
 plt.xlabel("残差 Residuals")  
 plt.title("Residual Distribution (LinearRegression)")  
 plt.tight\_layout()  
 resid\_path = os.path.join(args.out\_dir, "residuals.png")  
 plt.savefig(resid\_path, dpi=150)  
 plt.close()  
  
 # 保存指标为 CSV，便于在报告中引用  
 df = pd.DataFrame([  
 {"model": "LinearRegression", "mae": mae, "mse": mse, "r2": r2}  
 ])  
 metrics\_path = os.path.join(args.out\_dir, "metrics.csv")  
 df.to\_csv(metrics\_path, index=False, encoding="utf-8-sig")  
  
 # 6) 保存训练好的模型，便于后续加载与演示  
 print("[6/6] 保存模型与指标...")  
 import joblib # 与 sklearn 同源，常用来保存模型  
 model\_path = os.path.join(args.out\_dir, "linear\_model.joblib")  
 joblib.dump({  
 "model": model,  
 "feature\_names": feature\_names,  
 "note": "Simple LinearRegression trained on sklearn diabetes dataset"  
 }, model\_path)  
  
 print("输出完成：")  
 print(" - 指标:", metrics\_path)  
 print(" - 散点图:", scatter\_path)  
 print(" - 残差图:", resid\_path)  
 print(" - 模型:", model\_path)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

1. 运行（无需改代码）：

python exp1\_linear\_regression.py --out-dir outputs --test-size 0.2 --seed 42

1. 查看输出：

* 指标：outputs/metrics.csv
* 图像：outputs/pred\_scatter.png、outputs/residuals.png
* 模型：outputs/linear\_model.joblib

## 预期输出

* 控制台打印 MAE/MSE/R²，三者一起帮助判断模型是否合理。
* 两幅图：散点图应围绕 y=x 参考线分布；残差分布应较集中、近似对称更好。
* 已保存的 linear\_model.joblib 可用于课堂演示“加载即用”。

## 思考与讨论

1. 如果把“训练集更大、测试集更小”（如 9:1），R² 会更高吗？为什么需要留一部分数据做“考试”？
2. 当看到散点图中有少量点远离 y=x 参考线，你会如何形容这些点？在现实中它们可能意味着什么？
3. 如果我们不做任何“调参”，仅用最简单线性回归，你觉得它已经解决了哪些问题？哪些问题它可能无能为力？

## 常见错误与排错

* 未激活虚拟环境导致 ModuleNotFoundError：先执行 \.venv\Scripts\activate.bat。
* 图像没保存：确保已创建 outputs 目录，或在报错后重新运行脚本。
* 数值看不懂：从 MAE（平均误差的大小）入手直观比较，同时观察散点图与残差分布。

## 提交要求与评分标准

* 提交内容：PDF 实验报告 + outputs 目录（含 metrics.csv、两幅图、模型文件）。
* 命名规范：学号\_姓名\_实验一
* 评分 Rubric：准确性 40% / 完整性 30% / 表达 20% / 规范 10%