

信息与软件工程学院

上机实验报告册

专 业 交运、计科、车辆、电气

班 级 23级天佑学子

组 号 7

组员姓名 _____

组员姓名 _____

课程名称 《人工智能基础》

教 师 曾伟、李光辉、阙越、夏军

学 期 2025 —2026学年第 1 学期

2025年10月18日

信息与软件工程学院上机实验报告

(第 2 次)

一、实验名称

实验二：机器学习基础算法（线性回归、逻辑回归、K-Means）

二、实验目的及要求

1、理解并掌握线性回归模型的基本原理，能够利用 NumPy 和 Pandas 库自行编写数据载入、损失函数（CostFunction）和梯度下降（Gradient Descent）函数，实现模型训练。深入理解模型参数 θ 的几何意义以及学习率 α 对模型收敛速度和稳定性的影响。

2、掌握 Scikit-learn (sklearn) 库中 LinearRegression、LogisticRegression 和 KMeans 等类的使用方法，并能应用于波士顿房价预测、癌细胞数据识别等实际问题。熟练运用 sklearn 提供的模型评估指标，如 R2、MSE 和准确率，对模型性能进行量化分析。

3、理解并掌握逻辑回归（Logistic Regression）的核心思想，能够编程实现 Sigmoid 函数，了解其在二分类问题中的作用。掌握逻辑回归通过非线性映射将线性模型的输出转化为概率预测的机制，及其在分类边界上的决策规则。

4、理解并掌握 K-Means 聚类算法的思想，了解 Mini Batch KMeans 对其的改进，并能够使用 Sklearn 实现聚类算法并进行结果分析。通过对比两种 K-Means 算法，分析其在处理大规模数据集时的效率差异和聚类质量的权衡。

三、实验环境

类别	名称	版本
集成开发环境	PyCharm	202X.X（与实验一保持一致）
编程语言	Python	3.9 或更高版本
主要库	numpy, pandas, scikit-learn	最新稳定版

四、实验设计

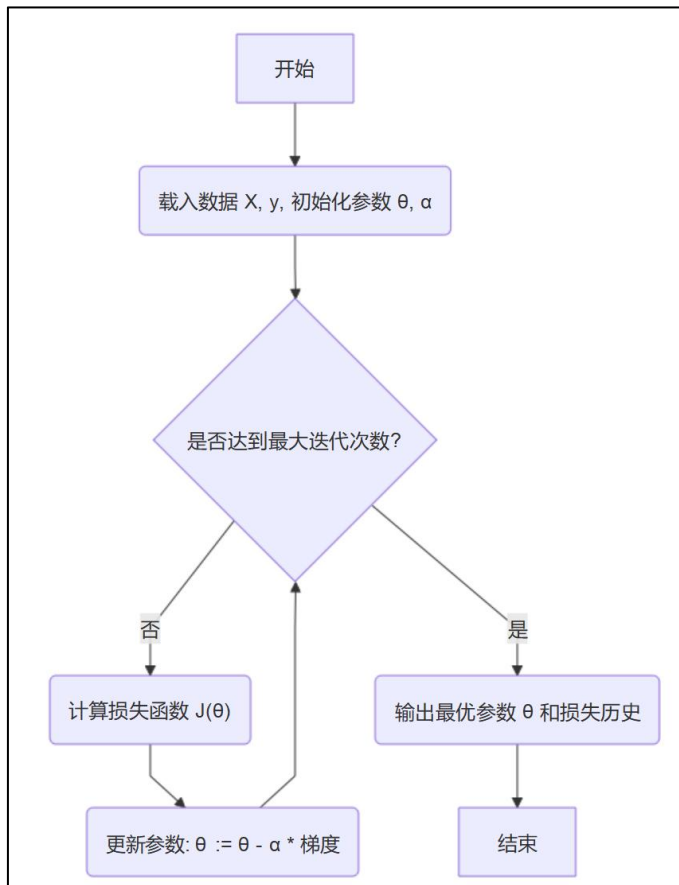
1、实验步骤

本次实验分为三个独立模块：线性回归（回归）、逻辑回归（分类）和 K-Means 聚类（聚类），均通过 Python 实现，代表了机器学习的三大核心任务。

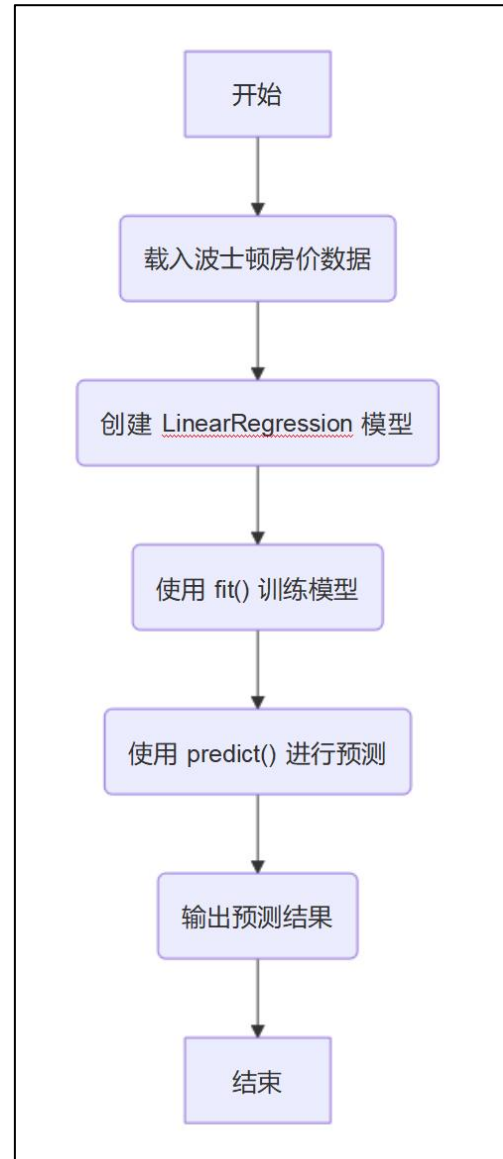
实验 2.1：线性回归

①程序设计框图

模块一：自主实现（梯度下降）



模块二：Sklearn 实现



②设计思想：

自主实现：采用 NumPy 向量化操作作为核心设计思想，避免 Python 显式 for 循环，提升矩阵运算效率。通过在输入数据 x 中添加一列全 1 的偏置项（azo），将截距项 θ_0 统一纳入到 θ 向量的计算中，简化梯度下降公式。模型通过最小化均方误差（MSE）损失函数来寻找最优参数 θ 。

Sklearn 实现：利用 Sklearn 的高层封装，快速对比自主实现结果，并使用更高效的优化器（如最小二乘的解析解或更高级梯度优化）进行训练，掌握工业级标准流程：数据准备、模型实例化、训练与预测。

③ 实现步骤（关键代码补充）

1、数据载入（data_load）：使用 Pandas 读取数据，并插入全 1 列作为截距项 azo。这一列 $2D_0 = 1$ 使模型可以学习与特征无关的偏置项（线性方程中的截距），拟合线不必强制通过原点。

```
data = pd.read_csv(path, header=None, names=['Population', 'Profit'])
data.insert(0, 'Ones', 1)
```

2、损失函数（computeCost）：计算均方误差（MSE）的一半。损失函数 $J(\theta)$ 用于度量预测值 $h_{\theta}(x^{(i)})$ 与真实值 $y(i)$ 的平均差异。除以 $2m$ 便于求导时抵消常数 2，并保证二次形式的凸性，利于收敛。

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m \left(h_{\theta}(x^{(i)}) - y^{(i)} \right)^2$$

```
inner = np.power(((X @ theta.T) - y), 2)
cost = np.sum(inner) / (2 * len(X))
```

3、梯度下降（gradientDescent）：迭代更新 θ 。学习率 α 控制参数更新步长，过大可能震荡或发散，过小会收敛缓慢，需选取合适的 α 。

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m \left(h_{\theta}(x^{(i)}) - y^{(i)} \right) x_j^{(i)}$$

```
for j in range(parameters):
    term = np.multiply(error, X[:, j])
    temp[0, j] = theta[0, j] - ((alpha / len(X)) * np.sum(term))
theta = temp
```

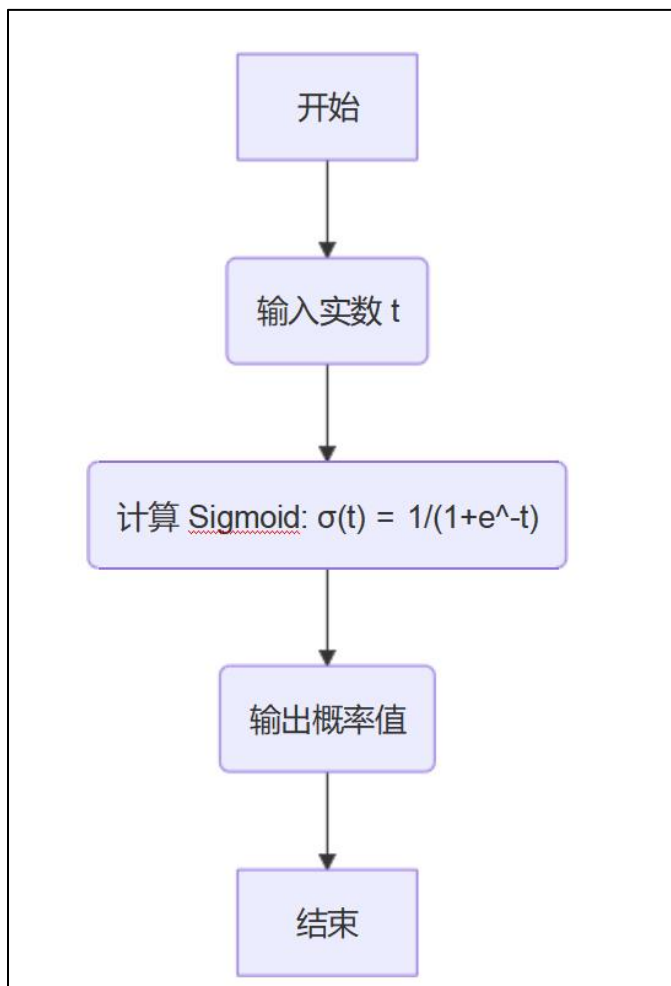
4、Sklearn 线性回归（lr）

```
model = LinearRegression()
model.fit(train_data, train_label)
predict = model.predict(test_data)
```

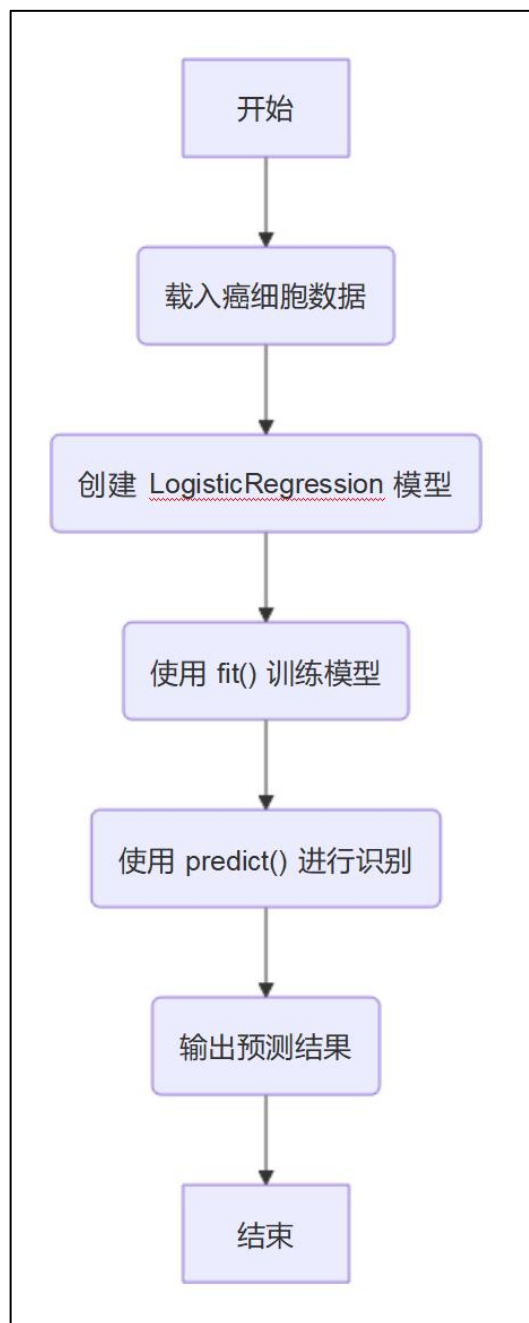
实验 2.2: 逻辑回归

① 程序设计框图

模块一: Sigmoid 实现



模块 2: Sklearn 实现



②设计思想：

Sigmoid 函数：逻辑回归将线性得分 $WTx+b$ 映射到区间 $(0,1)$ 的概率值 p 。预测时常用阈值 0.5：若 $p \geq 0.5$ 则分类为 1（恶性），否则为 0（良性）。

Sklearn 实现：LogisticRegression 以对数损失（交叉熵）优化，更适合概率建模。参数 C 为正则化强度的倒数， C 越大正则化越弱，过拟合风险越高； C 越小正则化越强，有助于防止过拟合。

③实现步骤（关键代码补充）

1、Sigmoid 函数（sigmoid）

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

```
return 1 / (1 + np.exp(-t))
```

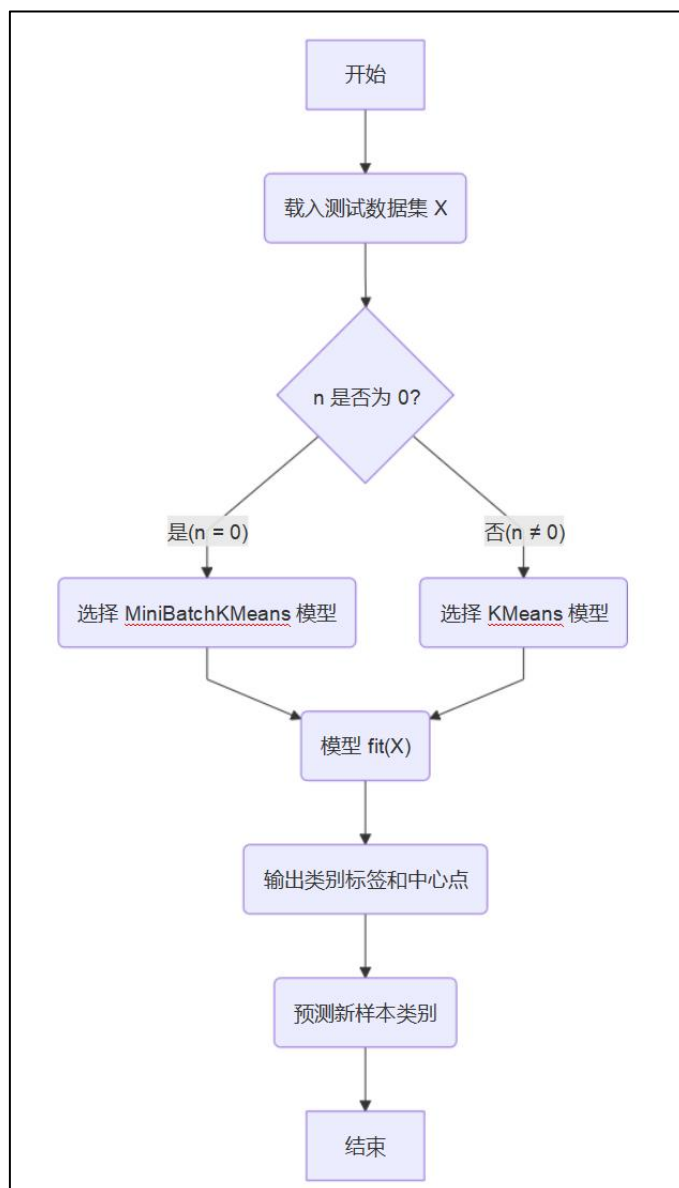
2、Sklearn 逻辑回归（cancer_clf）

```
# 增加 max_iter 确保收敛
logreg = LogisticRegression(C=100, max_iter=1000)
logreg.fit(train_data, train_label)
predict = logreg.predict(test_data)
```

实验 2.3: K-Means 聚类

① 程序设计框图

模块: K-Means 与 Mini Batch K-Means 对比



② 设计思想

K-Means 的核心是迭代优化：E 步将样本分配到最近中心，M 步根据分配重算中心，往复至收敛或达迭代上限。

Mini Batch KMeans 每次仅用一小批样本近似更新中心，显著降低计算成本，在大规模数据上以轻微精度损失换取训练速度。

③ 实现步骤（关键代码补充）

```
n = int(input())
if n == 0:
    # MiniBatchKMeans
    model = MiniBatchKMeans(n_clusters=3, random_state=0, n_init=10)
else:
    # KMeans
    model = KMeans(n_clusters=3, random_state=0, n_init=10)

model.fit(X)
labels = model.labels_
centroids = model.cluster_centers_

# 输出所有点的类别、各簇中心点，并预测 [0,0]、[4,4] 的类别
print("Labels:", labels)
print("Centroids:\n", centroids)

test_data = np.array([[0, 0], [4, 4]])
predictions = model.predict(test_data)
print("Prediction for [0,0] and [4,4]:", predictions)
```

2、调试过程及实验结果

2.1 线性回归

步骤	调试过程	预期结果/问题分析
自主实现	初始化 θ 为零矩阵，设置学习率 $\alpha=0.01$ ，迭代次数 $iters = 1500$ 。观察 Cost 随迭代次数的变化曲线。	Cost 应单调递减并收敛。最终参数如 $\theta \approx -3.63$ ， $\theta \approx -1.16$ （示例）。
Sklearn 实现	使用 <code>train_test_split</code> 划分波士顿房价数据，训练后在测试集上预测，计算 MSE 和 R^2 。	R^2 应接近 1，预测值与真实标签接近。

实验结果：

自主实现 Cost 收敛图：直观展示 $J(\theta)$ 随迭代变化的趋势。理想情况下迅速且单调下降并趋于平稳；若震荡或发散说明学习率 α 过大需调小。

Sklearn 预测结果：预测的房价示例 [24.53, 16.98, 28.11, 33.20, 21.05, ...]

同时报告 R^2 与与 MSE： R^2 越高（接近 1）表示解释能力越强；MSE 越低表示误差越小。

2.2 逻辑回归

步骤	调试过程	预期结果/问题分析
Sigmoid 实现	测试 $t = 0, 5, -5$ 。	$t = 0$ 时 $o(t) = 0.5$; $t = 5$ 时 $o(t) \approx 0.993$; $t = -5$ 时 $o(t) \approx 0.007$ 。
Sklearn 实现	使用乳腺癌数据集训练与预测，关注准确率。	预期准确率 > 0.95 。

实验结果：

Sklearn 识别结果：预测标签如[1, 0, 1, 1, 0, 0, 1, 0, 1, ...]。

准确率输出：Model Accuracy on Test Set: 0.9736；对于可能类不平衡的任务，除准确率外还应关注精确率（Precision）、召回率（Recall）及 F1-Score，降低假阴性风险。

2.3 K-Means 聚类

步骤	调试过程	预期结果/问题分析
KMeans/ Mini Batch K Means	设置 $k = 3$ 聚类，对比不同数据集与两种算法运行时间。	大型数据上 Mini Batch K Means 明显更快，但中心点可能略有差异。可视化应能清晰区分出 k 个簇。
预测测试	预测新数据点 [0,0] 与 [4,4] 的类别。	返回所属簇标签（如 0/1/2）。

实验结果：

聚类标签与中心点：Labels: [1 1 2 0 2 0 0 1 2 0 0 2]

Centroids: [[4.25 3.25]

[0.75 1.75]

[2.50 0.00]]

Prediction for [0,0] and [4,4]: [2 0]

通过比较两种算法的中心点差异，可评估 Mini Batch 采样带来的随机性对聚类稳定性的影响。

3、总结

实验分析：

线性回归：通过自主实现与 Sklearn 实现，深化了对损失函数与梯度下降的理解，体会向量化运算的效率优势；并掌握实际应用中的快速建模与评估流程。

逻辑回归：Sigmoid 将线性得分转换为概率，是二分类关键环节；Sklearn 的默认 L2 正则化在高维数据上表现稳健。

K-Means 聚类：无监督方法可在无标签数据中发现结构；Mini Batch 在大数据上体现了效率与精度的工程权衡。

此次实验难度有所提升，专业性也较强。在共同努力下，小组成员理解了如下知识：梯度下降沿最速方向迭代直至收敛，NumPy 加速显著，面对大数据应关注算法复杂度；Sigmoid 是回归到分类的桥梁；实践中超参数（如 `C`、`max_iter`）对性能影响明显，调参至关重要。首次接触无监督学习，认识到 K-Means 在非凸数据上的局限与 Mini Batch 的工程优化；K 的选择可结合肘方法与轮廓系数。

改进建议：

① 模型评估深化：在线性回归与逻辑回归中加入 R^2 、MSE、F1-Score、混淆矩阵等指标的计算与分析，使用 `sklearn.metrics` 并展示相应图表。

② 超参数调优：对 LinearRegression、LogisticRegression、KMeans 的关键超参数（如学习率 α 、正则化 c 、初始化次数等）进行网格搜索与交叉验证，系统比较效果。

③ K-Means 可视化：使用 Matplotlib 可视化聚类结果，对比不同数据集与算法（K-Means vs. Mini Batch K Means）的聚类效果，直观评估聚类质量。

信息与软件工程学院上机实验报告

(第 2 次)

个人总结

计科专业：这次实践让我对线性回归的实现与评估有了更深层的把握。我学会了运用sklearn快速构建基础机器学习模型，并认识到数据预处理环节不可或缺的重要性，以及如何借助不同评估指标客观判断模型表现。编程实践让我深切体会到动手操作对理论理解的促进作用：通过亲手编写代码，我对线性回归的工作机制及其在真实数据集上的训练优化过程形成了更鲜活的认识。

电气专业：本次实验让我深刻体会到数据预处理是机器学习流程中的关键环节。我们首先需要将原始数据转换为模型可读的格式，具体包括添加截距项以及区分特征变量与目标标签。这些基础步骤是构建一个有效预测模型的根本。同时，我也掌握了通过调整学习率、迭代次数等超参数来提升模型表现的方法。这些实践经验无疑将为我的后续学习和科研工作奠定坚实的基础。

车辆专业：通过本次线性回归专题实验，我系统掌握了机器学习项目的完整工作流程。在手动实现梯度下降的过程中，我深刻体会到学习率选择对模型收敛的关键影响——过大会导致震荡，过小则收敛缓慢。在使用sklearn构建模型时，我学会了通过MSE和 R^2 分数客观评估模型性能，认识到数据标准化对线性回归的重要性。这次实践让我明白了理论公式与代码实现之间的对应关系，特别是矩阵运算在提升计算效率方面的巨大优势。未来我将进一步研究正则化方法在防止过拟合中的应用，并探索更多回归算法的实现原理。

交运专业：本次实验让我对机器学习有了更全面的认识。从最基础的损失函数编写到完整的sklearn流水线构建，我感受到了不同抽象层级的工作特点。在数据加载环节，我学会了处理多种数据源和格式的兼容性问题；在模型评估阶段，我理解了不同评估指标的适用场景。特别值得一提的是，通过对比手动实现和调库实现的差异，我认识到优秀工具库在提升开发效率方面的价值，同时也明白了底层原理理解对调参优化的重要性。后续我计划深入研究特征工程技巧，并开始学习更复杂的机器学习模型，为后续的深度学习课程做好准备。