**中山大学计算机学院**

**人工智能**

**本科生实验报告**

课程名称：Artificial Intelligence

|  |  |  |  |
| --- | --- | --- | --- |
| 学号 | **23336179** | 姓名 | **马福泉** |

# 实验题目

感知机算法

1. 房价预测任务

data.csv数据集包含五个属性，共10000条数据，其中 longitude和 latitude表示房子经纬度，housing\_age表示房子年龄，homeowner\_income表示房主的收入（单位：十万元）， house\_price表示房子的价格。请根据数据集 data.csv中四个属性 longitude、latitude、housing\_age、homeowner\_income，利用感知机算法预测房价 house\_price，并画出数据可视化图、loss曲线图。

1. 提示：最后提交的代码只需包含性能最好的实现方法和参数设置. 只需提交一个代码文件, 请不要提交其他文件.本次作业可以使用 numpy库、matplotlib库以及python标准库.数据集可以在Github上下载。

# 实验内容

1. 算法原理
2. 数据加载与预处理
3. 数据加载 - load\_data 函数：使用Python的csv模块读取文件，跳过表头，将每行数据转换为浮点数，最后返回NumPy数组
4. 异常值处理 - detect\_and\_remove\_outliers 函数

IQR方法：基于箱线图原理，计算每个特征的四分位数范围，将超出(Q1-1.5IQR)和(Q3+1.5IQR)的样本视为异常值

Z-score方法：计算每个特征值的Z分数，将绝对Z分数大于阈值的样本视为异常值

返回：清洗后的特征数据、目标变量及标识异常值的布尔掩码

1. 数据标准化 - normalize\_data 函数

对每个特征计算均值和标准差，然后使用公式(X-mean)/std进行标准化

1. 地理特征转换

transform\_geo\_features 函数

1. 实现原理：使用K-means聚类将地理位置分为5个区域
2. 计算每个样本点到各区域中心的距离作为新特征
3. 将得到的5个距离特征与原有的房龄、收入特征合并

这段 Python 代码实现了一个用于回归任务（预测房价）的多层感知机 (MLP) 模型，并包含了详细的数据预处理步骤。

1. MLP 模型构建:

定义了一个 `MLP` 类。

1. 初始化:根据指定的网络层级大小 (`layer\_sizes`) 初始化权重 (使用 He 初始化) 和偏置。
2. 前向传播 (`forward`):实现了混合激活函数策略：根据层的位置选择不同的激活函数（例如，第一隐藏层用 tanh，第二隐藏层用 ReLU，输出层用 linear）。

计算每一层的输出。

1. 后向传播 (`backward`):计算输出层误差。

根据链式法则和激活函数的导数，逐层反向计算梯度。

1. 使用梯度下降法更新权重和偏置。
2. 模型训练与评估:

定义 MLP 的网络结构（例如，输入层-20节点隐层-10节点隐层-输出层）。

实例化 `MLP` 模型，设置学习率和迭代次数。

使用训练集训练模型。

使用测试集评估模型性能，计算标准化 MSE、原始尺度 MSE 和 R² (决定系数)。

1. 可视化部分

数据清理可视化：对比清理前后各特征与房价的关系

地理聚类可视化：展示K-means聚类的区域划分

损失曲线可视化：展示模型训练过程中损失的变化

MSE比较可视化：比较不同参数组合的模型性能

预测结果可视化：展示真实值与预测值的对比，以及特征与房价的关系

特征重要性可视化：展示各特征对预测结果的影响程度

1. 关键代码展示
2. **数据读取与处理**



1. **经纬度联合处理**



1. **应用激活函数**



1. 创新点&优化
2. 数据预处理（代码见上文关键代码展示）
3. 多种异常值处理方法的整合：实现了两种常见的异常值检测算法（IQR和Z-score）。
4. 特征与目标变量的联合异常值处理：任何维度出现异常都会使整个样本被标记，确保数据质量
5. 处理前后对比可视化：直接通过图形对比展示清理前后的数据分布变化：用不同颜色突出显示被识别的异常点
6. 精细的数据标准化设计
7. 经纬度联合考量（代码见上文关键代码展示）

用K-means聚类将地理位置分为5个区域

计算每个样本点到各区域中心的距离作为新特征

将得到的5个距离特征与原有的房龄、收入特征合并

1. 不同参数组合的考量与选择

尝试不同的学习率和迭代次数组合

训练多个模型并评估其性能

可视化不同参数下的损失曲线

模型评估

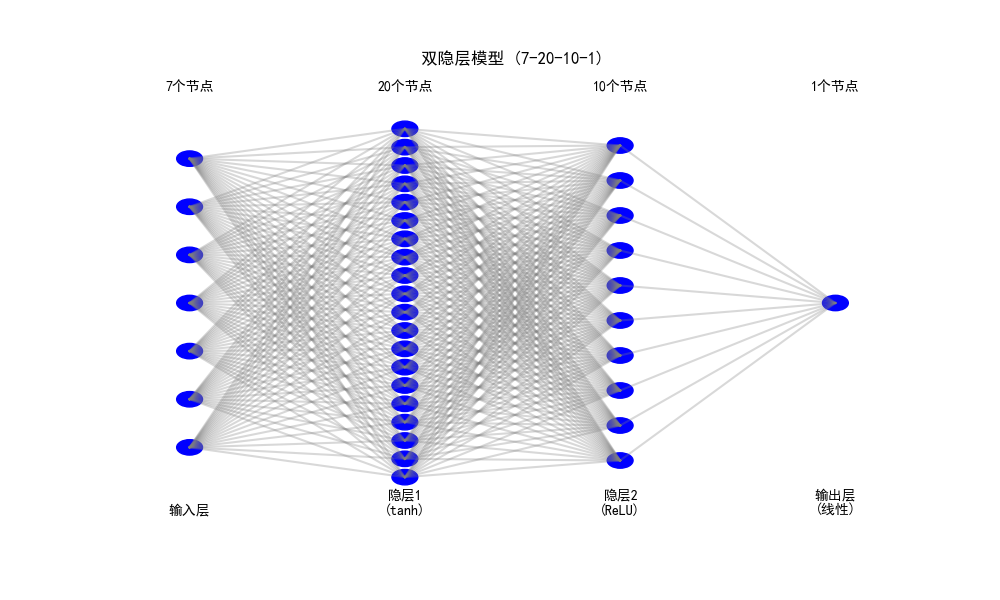
1. 混合使用多种激活函数策略

• 第一隐藏层：使用 tanh 激活函数，输出范围为 [-1, 1]，在 0 附近具有较强梯度

• 中间隐藏层：使用 ReLU 激活函数，解决深层网络的梯度消失问题

• 输出层：不另外使用激活函数，使用线性激活，适合回归任务的任意范围输出。

结合了不同激活函数的优势，充分发挥了 tanh 在输入处理和 ReLU 在中间层特征提取方面的各自优点。



1. 完整的可视化过程

数据清理可视化：对比清理前后各特征与房价的关系

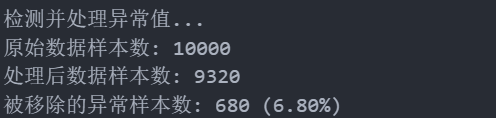
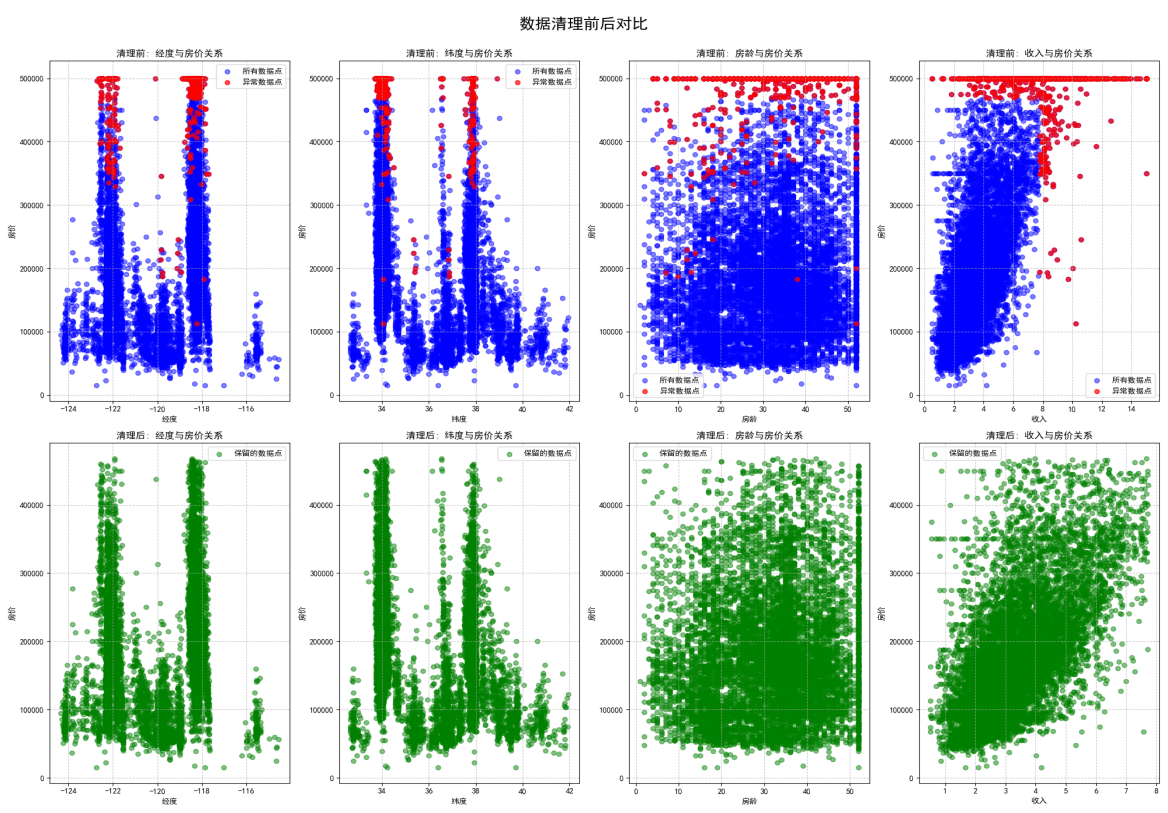
地理聚类可视化：展示K-means聚类的区域划分

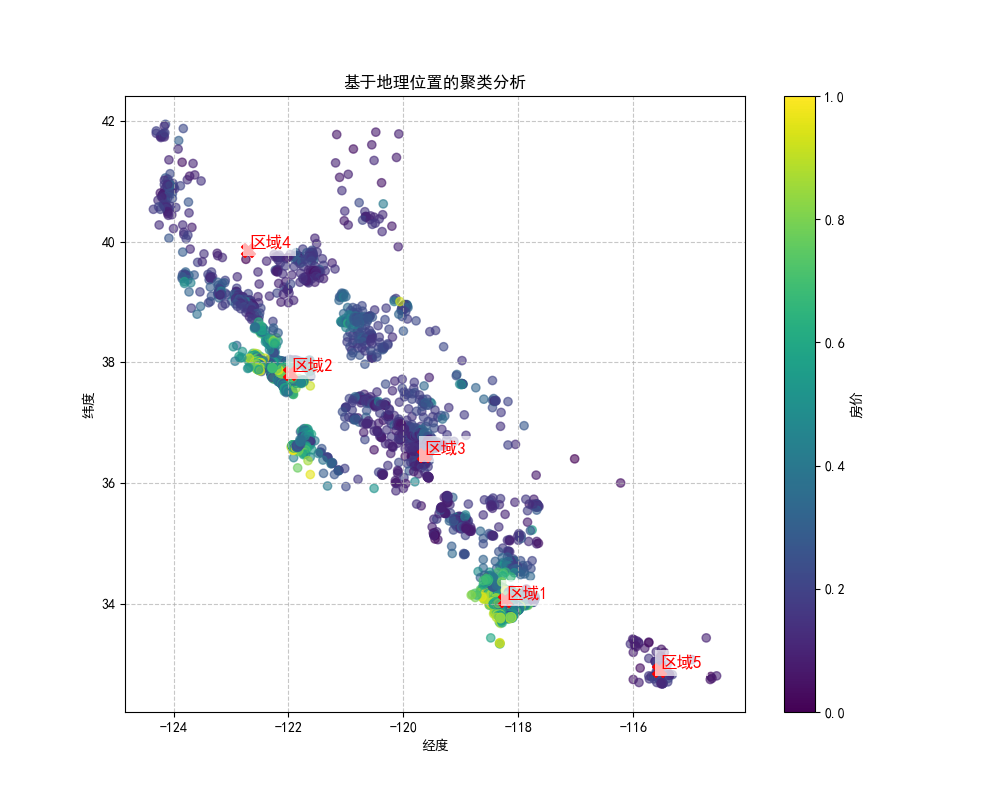
损失曲线可视化：展示模型训练过程中损失的变化

预测结果可视化：展示真实值与预测值的对比，以及特征与房价的关系

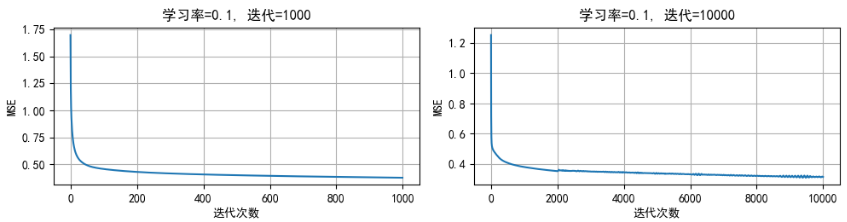
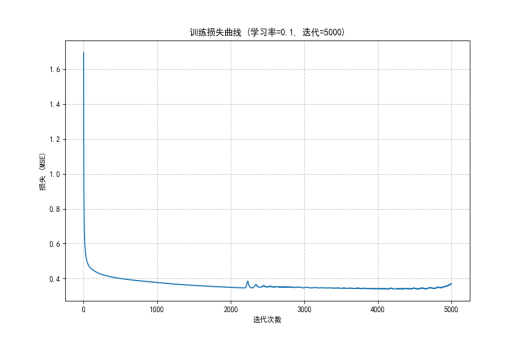
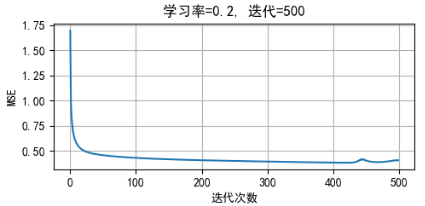
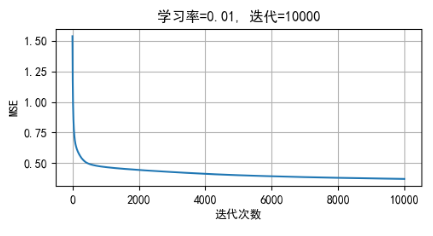
特征重要性可视化：展示各特征对预测结果的影响程度

# 实验结果及分析

1. 实验结果展示示例
2. 数据预处理结果（data\_cleaning\_comparison.png）  
     
   如上图，采用IQR方法去除红色数据。
3. **经纬度处理结果（geo\_clustering.png）**

  
如图所示，采用Kmeans算法，得到五个区域。

1. 不同参数组合的比较选择（过程代码已舍去）

选择lr=0.1,ep=5000的参数组合,损失值在1000次迭代后稳定在0.4附近，表明模型已完成主要学习过程,稳定性后期小幅波动（±0.005）属于正常随机梯度下降现象，未出现剧烈震荡.

1. 学习率（lr）=0.1：

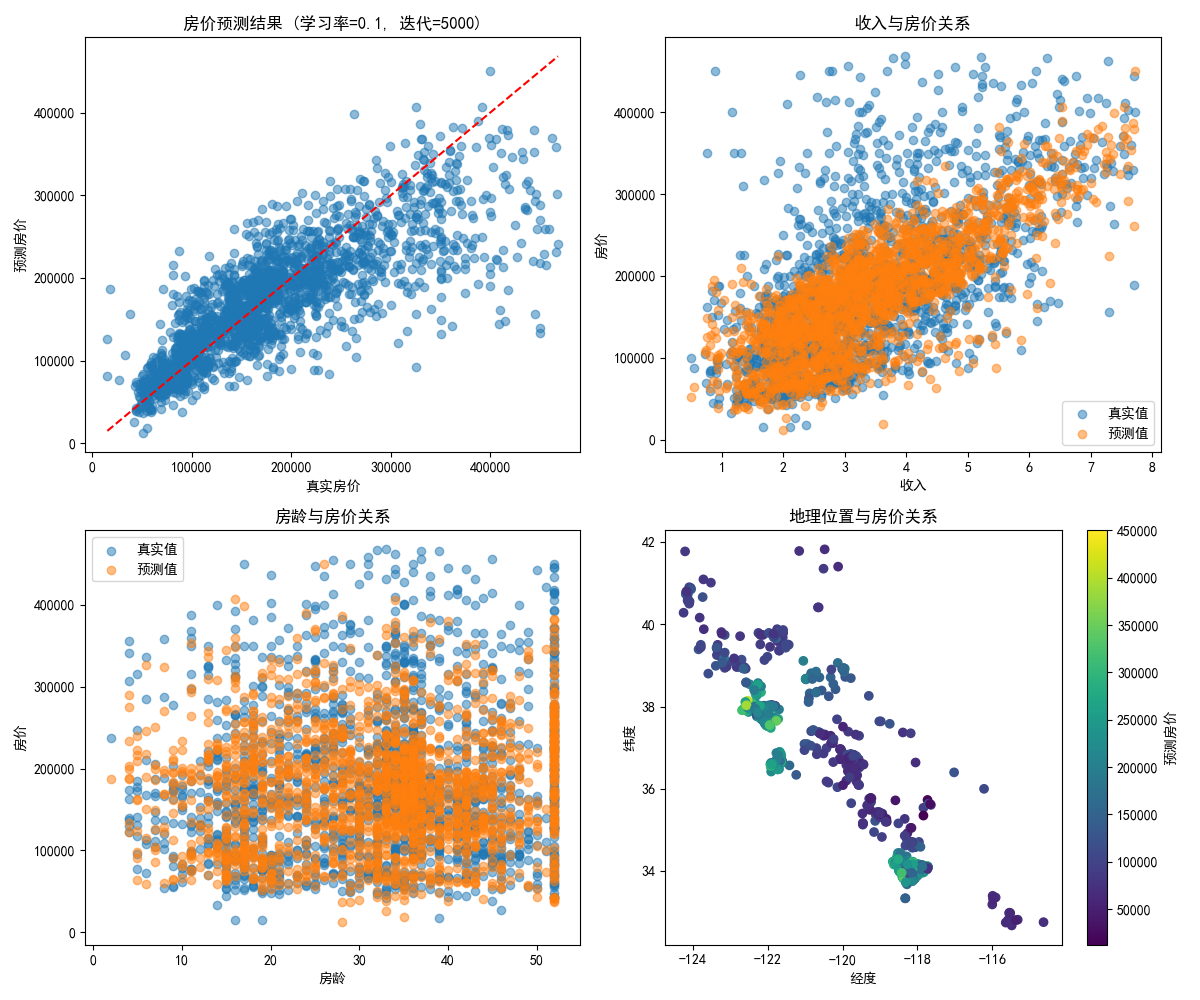
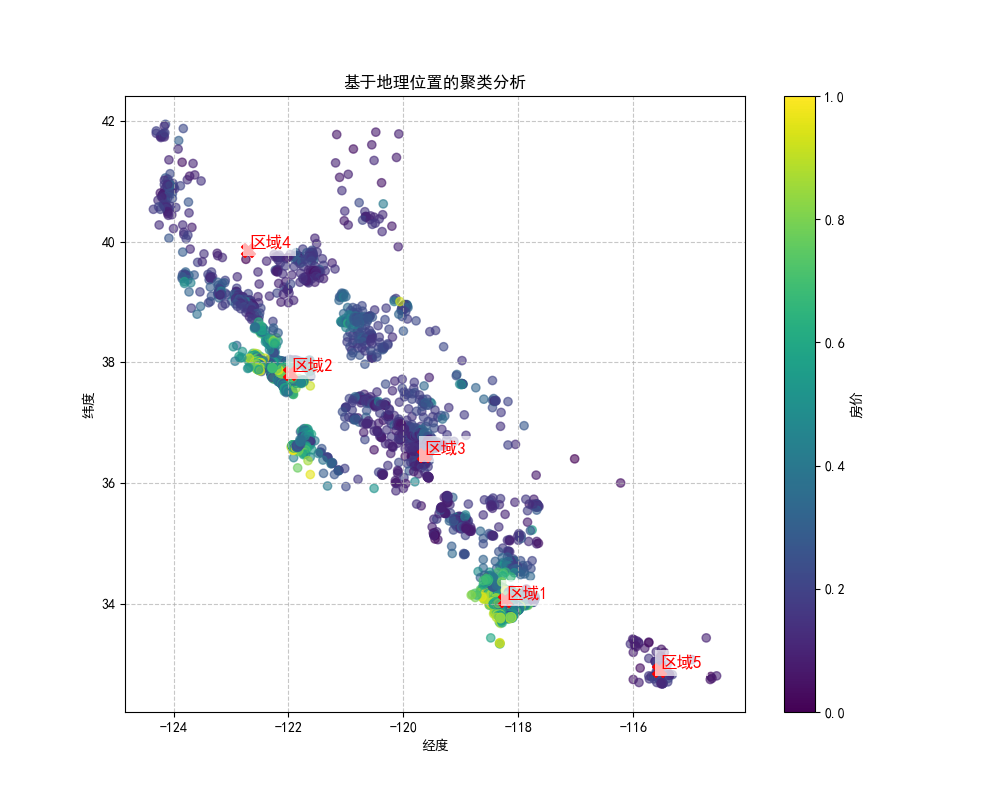
当学习率为0.2时（左列三图），误差曲线初期下降较快，但后期波动明显且收敛平缓，说明学习率过大可能导致模型在最优解附近震荡，难以稳定收敛。

而学习率0.1时（右列三图），误差曲线下降更平滑，尤其在迭代10000次后趋于稳定，表明较小的学习率能更精准地逼近最优解，避免过冲或震荡。

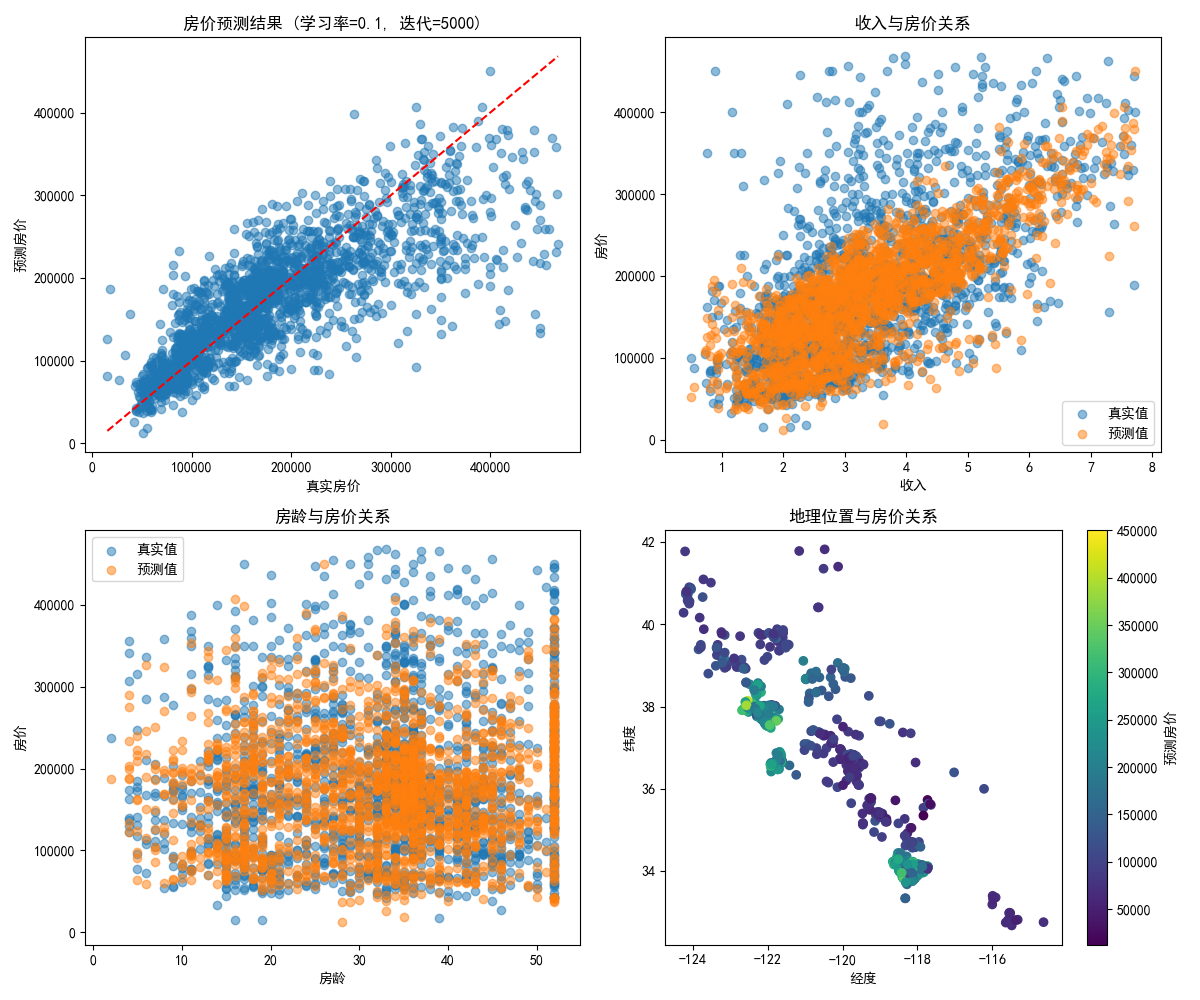
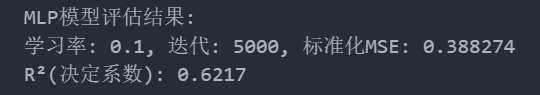
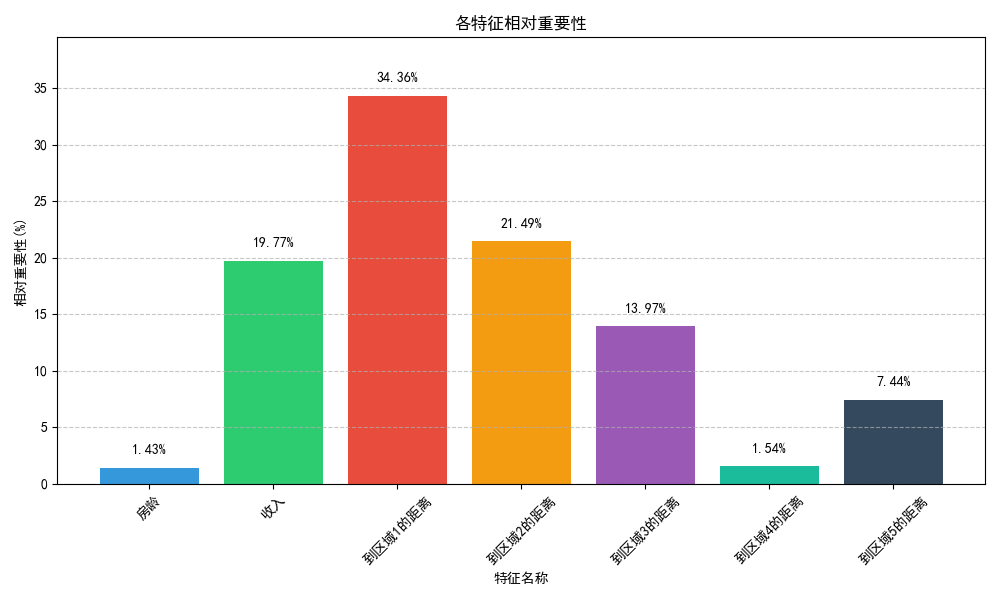
学习率0.1既不会因过大（如0.2）而跳过最优解，也不会因过小（如0.01）而需要极高迭代次数，在效率和精度间取得了平衡。

1. 迭代次数=5000 的合理性​

在lr=0.1时，迭代5000次的误差曲线（中下图）已接近最低点，继续增加到10000次（右下图）误差仅小幅下降但计算成本翻倍，而且出现数据不稳定现象。相比之下，5000次是保证充分收敛的临界点。

1. 最佳参数的预测结果  
   

（左图为真是数据，右图为预测数据）

各变量维度预测结果与真实数据对比如上图，预测结果与实际数据较为吻合，以及可视化的各个特征重要值。

1. 评测指标展示及分析
2. 模型性能评估：标准化MSE=0.388274​，R²=0.6217（决定系数）​​

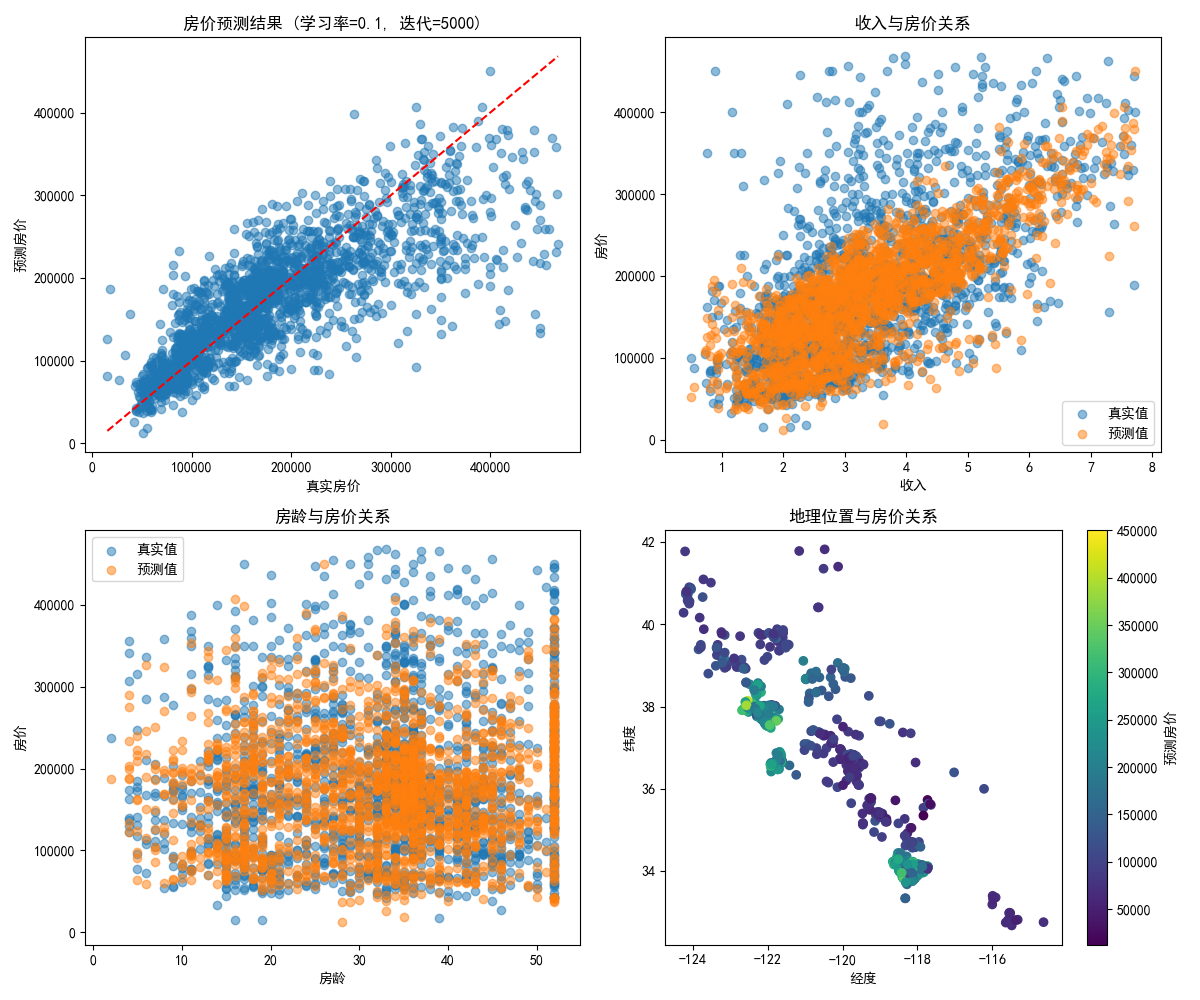
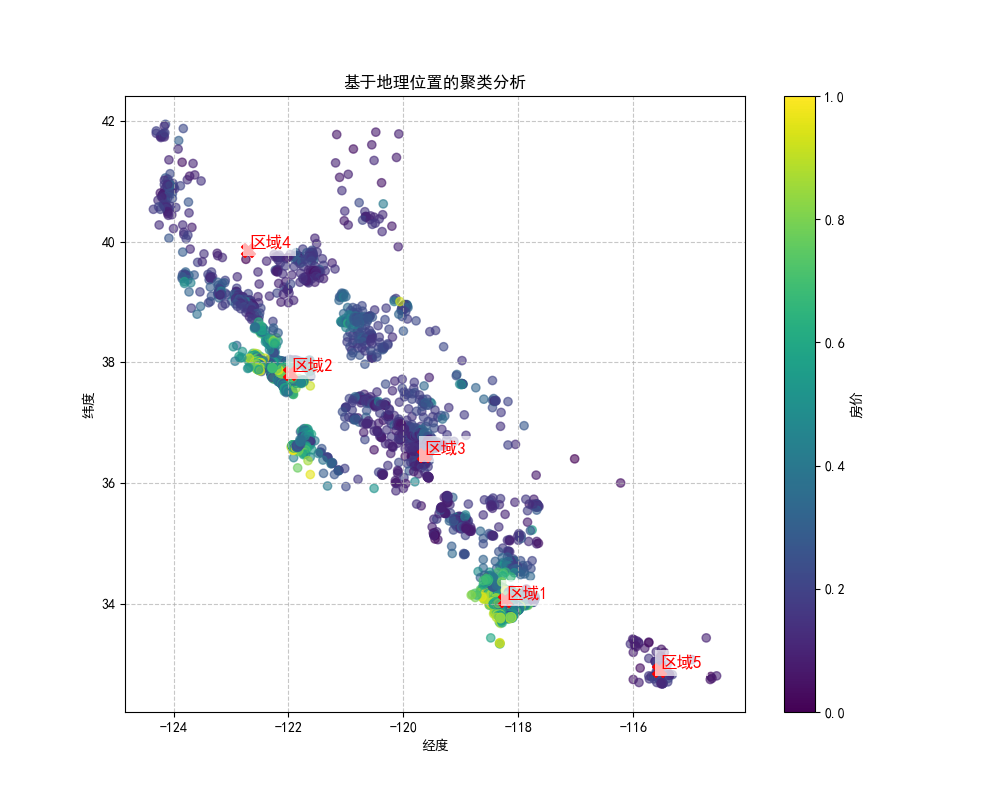
解释力​：模型能够解释目标变量（房价）62.17%的方差，剩余约38%的波动未被捕捉。

在房价预测问题中，R²通常期望达到0.5~0.8（中等以上），考虑到房价还有其他因素影响，且采用的只是简单的线性预测模型，当前结果合理但仍有提升空间。则当前模型有一定预测价值，但需进一步优化。

1. 参数合理性分析​​

学习率：0.1是中等偏大的学习率，适合快速收敛。

迭代次数：5000次迭代可能已使模型接近收敛，但需检查训练曲线：训练损失和验证损失均平稳，说明迭代足够；

1. 变量重要性分析​  
   
2. 到区域1的距离（34.36%）：绝对重要性最高（0.9628），可能反映核心区位价值（如市中心、商业区）。
3. 收入（19.77%）：重要性次之（0.5538），符合经济常识（收入高的人群购房能力更强）。
4. 到区域2，3的距离（合计35.46%）：可能与区域1形成互补（如次级商圈或居住区），需分析这些区域的实际功能。
5. 到区域5的距离（7.44%）：贡献较低但不可忽略，可能对应郊区或特定设施（如公园）。
6. 房龄（1.43%）和到区域4的距离（1.54%）：影响微弱，但需验证是否因数据分布或模型缺陷导致低估（如房龄与房价实际为非线性关系）。

（4）后续改进建议​​

1. 升级模型​：尝试非线性模型（如XGBoost、随机森林），自动捕捉变量间复杂关系。
2. 数据增强​

​补充特征​：加入房屋面积、卧室数量、周边设施（地铁、学校）等关键因子。

​数据变换​：对“距离”类变量尝试分箱或对数变换，解决非线性问题。

# 参考资料

**（1）部分 debug，优化建议参考了 ai 大模型的建议。**

**（2）阅读参考了以下文档：  
课程实验指导文档**

[多层感知机（MLP）简介-CSDN博客](https://blog.csdn.net/fg13821267836/article/details/93405572)

[MLP神经网络：多层感知机实现波士顿房价预测\_mlp波士顿-CSDN博客](https://blog.csdn.net/weixin_46287760/article/details/130933840)  
[【人工智能】保姆级波士顿房价预测-CSDN博客](https://blog.csdn.net/weixin_63669071/article/details/137357142)  
[python机器学习 波士顿房价预测 详细教程 数据集+源码+结果图+远程部署\_波士顿房价数据集-CSDN博客](https://blog.csdn.net/m0_68036862/article/details/139058692)  
[使用Python和Numpy构建神经网络模型——波士顿房价预测案例 - 知乎](https://zhuanlan.zhihu.com/p/131342575)