自然班级	天佑计算机 2023-1	姓名		小班		小班序号	31	成绩	



上机实验报告

题　　目：利用深度神经网络进行房价预测

学院（系）：　信息与软件工程学院

年级专业：　2023 级计算机科学与技术

学　　号：_____　　_____

姓　　名：_____　　_____

上课教师：　黄晓辉

信息与软件工程学院上机实验报告

（第 2 次）

一、实验名称

利用深度神经网络进行房价预测

二、实验目的及要求

1. 掌握深度神经网络在回归任务中的基本实现方法。
2. 掌握表格数据的预处理流程，包括缺失值处理、类别编码与数值标准化。
3. 理解 K 折交叉验证、对数误差评价指标及 Kaggle 提交流程。

三、实验环境

1. Python 3
2. Anaconda
3. PyTorch

四、实验内容

1. 参照上课例子完成 California 房价预测代码
2. 提交 Kaggle 评价自己的算法结果。

五、实验设计内容要求

1. 实验步骤

本实验的目标是使用神经网络完成加州房价预测。原始数据同时包含数值字段、类别字段、日期字符串以及房源简介等混合类型信息，因此程序不能直接将原始表输入模型，而必须先进行特征工程和统一的数值化处理。整体实现遵循“读取数据 → 构造特征 → 建立预处理器 → K 折训练 → 生成提交结果”的顺序展开。

1.1 设计思想

本实验采用较为基础的多层感知机模型，重点在于数据预处理和验证流程的规范性。

程序首先对原始数据进行特征构造。对于日期列，提取年份、月份以及挂牌时间与上次成交时间的间隔；对于文本列，保留简介长度这一简单统计特征；对于面积、价格和税额等字段，构造单位面积挂牌价和税率等比值特征。随后删除不适合直接进入模型的编号列、地址列、原始长文本以及已经拆分完成的日期字符串列。

在预处理阶段，程序自动区分数值列与类别列。数值列采用“中位数填补 + 标准化”的方式处理，类别列采用“缺失值填补 + One-Hot 编码”的方式处理，并通过设置低频类别阈值将罕见类别合并，从而避免维度过度膨胀。模型训练阶段使用 5 折交叉验证，在每一折中仅用训练折拟合预处理器，以避免信息泄漏。

1.2 实现流程

实验整体流程如图所示。

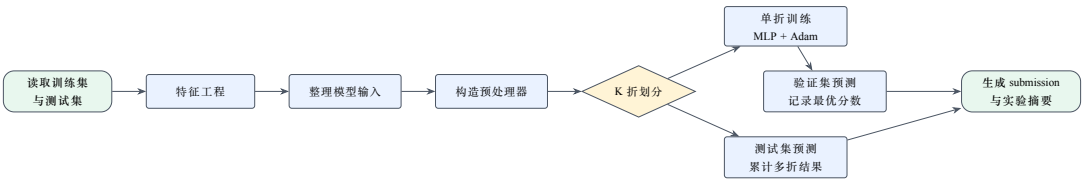


图 1: 实验二程序流程图

1.3 主要实现过程

程序的核心实现可以概括为以下几个步骤。

1. 使用 `pandas.read_csv()` 读取训练数据与测试数据，对应文件分别为 `california_train.csv` 和 `california_test.csv`。
2. 在 `add_features()` 中完成特征构造，包括卧室数量提取、日期拆分、房龄、简介长度、单位面积挂牌价和税率等。
3. 在 `build_model_input()` 中删除 `Id`、`Address`、`Summary`、原始日期列以及训练集中的标签列 `Sold Price`。

4. 在 `build_preprocessor()` 中建立预处理器：

   (a) 数值列：中位数填补缺失值，再做标准化；

   (b) 类别列：使用 `Missing` 填补缺失值，再做 One-Hot 编码；

   (c) 通过 `min_frequency=100` 合并低频类别。

5. 在 `make_fold_data()` 中，将训练折、验证折和测试集统一转换为数值矩阵，并将标签做 $\log(1+y)$ 变换。

6. 使用多层感知机作为回归模型，其结构为“输入层 $\to 256 \to 64 \to 1$”，激活函数为 ReLU，并在第一隐藏层后加入 `Dropout(0.2)`。

7. 使用 Adam 优化器和均方误差损失函数进行训练，并采用早停策略保存单折验证集分数最优的模型参数。

8. 在 `run_k_fold()` 中执行 5 折交叉验证，记录每一折验证分数，累加测试集预测结果，并最终输出平均预测。

本实验采用对数空间下的 RMSE 作为主要评价指标。设真实房价为 y_i，预测房价为 $\hat{y}_i$，则对数 RMSE 定义为：

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\log(1+\hat{y}_i) - \log(1+y_i)\right)^2}$$

由于程序中模型直接预测的是 $\log(1+y)$，因此实际计算时可写为：

$$\text{RMSE}_{\log} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(pred_i - target_i)^2}$$

其中 $pred_i$ 表示模型输出的对数房价，$target_i$ 表示真实房价的对数值。之所以采用该指标，是因为房价分布具有明显长尾特征，对标签做对数变换后，训练过程更加稳定，同时也与 Kaggle 评测方式保持一致。

设第 k 折对应测试集上的预测结果为 $\hat{y}_{test}^{(k)}$，则最终提交时使用 5 折平均结果：

$$\hat{y}_{test} = \frac{1}{K}\sum_{k=1}^{K}\hat{y}_{test}^{(k)}, \quad K = 5$$

2. 调试过程及实验结果

2.1 调试重点

实验调试过程中，主要集中在以下几个方面。

第一，原始数据中的字段类型较为复杂。部分列本应表示数值，但实际同时混有文本信息，例如 Bedrooms 字段既可能出现具体数量，也可能出现文字描述。为此，程序单独从该列中提取可解析的数值，形成 Bedroom count 新特征，同时保留原始列用于类别编码。

第二，类别字段数量较多，且部分字段具有较高基数。如果直接对全部类别值进行完整展开，会造成特征维度迅速增长，既增加计算量，也容易引入稀疏噪声。针对这一问题，程序删除了地址、原始简介等不适合直接编码的字段，并通过 OneHotEncoder(handle_unknown="infrequent_if_exist", min_frequency=100) 自动合并低频类别，有效控制了编码后的维度规模。

第三，多个连续值字段存在明显偏态分布，例如地块面积、室内面积、评估价、税额和挂牌价等。如果直接以原始尺度进行训练，梯度波动较大，模型收敛不够稳定。因此程序对这些高偏态字段以及标签 Sold Price 使用 $\log(1+x)$ 变换，以减轻长尾分布带来的影响。

第四，交叉验证过程中必须避免信息泄漏。若在整份训练集上拟合预处理器，再去处理某一折的验证集，则验证集信息已经间接参与训练。程序在 make_fold_data() 中严格限制预处理器只在当前训练折上 fit，再对验证折和测试集执行相同的 transform，从而保证验证结果具有参考意义。

第五，模型训练后期可能出现验证误差不再下降的情况。为减少无效训练，程序设置了 patience=5 的早停策略。当验证集分数连续 5 轮未改善时，训练提前结束，并恢复到验证集表现最好的模型状态。

2.2 实验结果

本实验的主要训练配置如下表所示。

表 1: 模型训练参数

参数项	取值
随机种子	42
K 折数	5
最大训练轮数	25
早停轮数	5
Batch Size	1024
学习率	0.002
Weight Decay	0.0001
低频类别阈值	100

5 折交叉验证结果如下表所示。

表 2: 5 折验证集 log-rmse 结果

折数	验证集 log-rmse
Fold 1	0.23275
Fold 2	0.24392
Fold 3	0.27228
Fold 4	0.22840
Fold 5	0.23461
平均值	0.24239

整体 OOF 分数为:

$$\text{OOF log-rmse} = 0.24291$$

各折验证结果可视化如图所示。

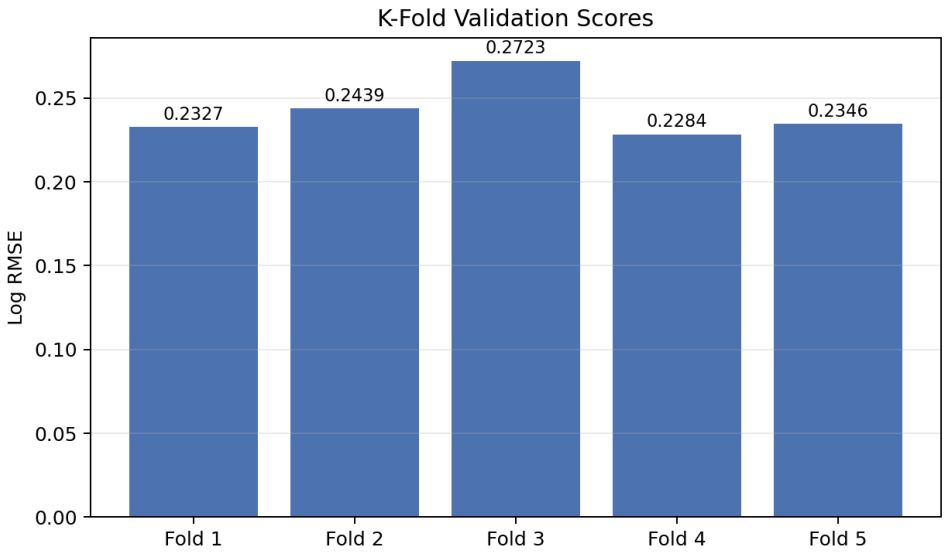


图 2: 5 折交叉验证分数统计图

此外，程序利用 Kaggle API 提交了 `submission.csv`，得到的线上评测结果如下图所示。

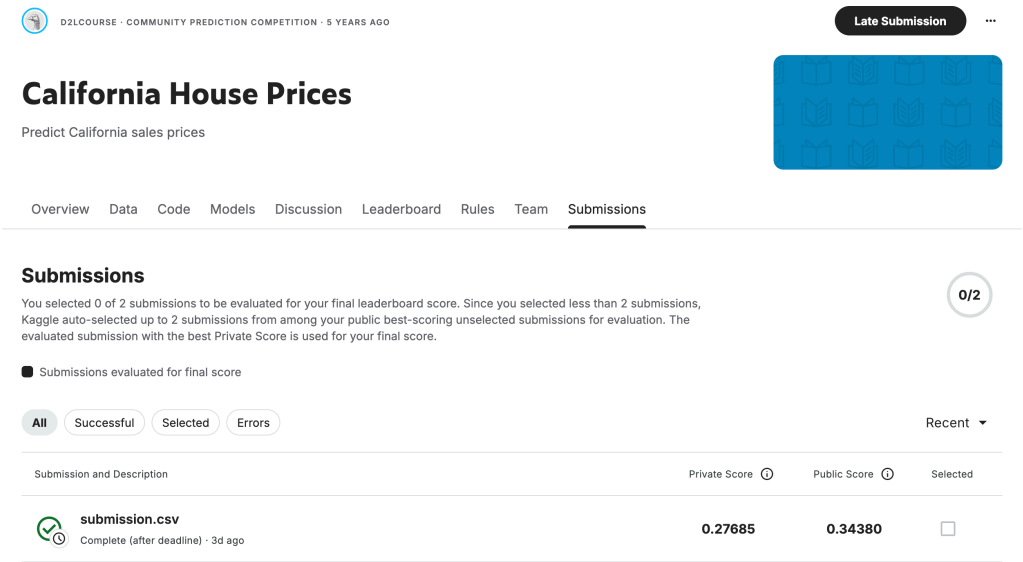


图 3: Kaggle 提交结果

2.3 结果分析

从交叉验证结果可以看出，5 折分数整体较为接近，说明模型在不同数据划分上的表现较稳定。其中第 3 折的误差略高于其他折，说明这一折的验证样本分布相对更复杂，但整体波动幅度不大，因此训练结果仍具有较好的可信度。

本地 5 折平均 log-rmse 为 0.24239，OOF 分数为 0.24291，两者十分接近，说明交叉验证评估较为稳定，没有出现明显偏离。Kaggle 公榜得分为 0.34380，私榜得分为 0.27685，与本地 OOF 分数相对接近，说明当前验证方式对线上表现具有一定参考价值。

从模型复杂度看，本实验所采用的网络结构仍属于较基础的多层感知机，没有使用特别复杂的深层网络、特征交叉网络或模型集成方法。实验结果表明，在表格回归问题中，只要预处理过程合理、特征工程有效、验证流程规范，即使使用较基础的神经网络模型，也能够取得较好的预测效果。

3. **总结**

通过本次实验，我对神经网络在表格回归任务中的完整实现流程有了较系统的认识。与实验一相比，本实验直接处理真实房价数据，数据规模更大、字段类型更复杂，因此也更能体现数据预处理的重要性。实验结果表明，在该类任务中，缺失值处理、类别编码、数值尺度变换和交叉验证策略都会直接影响模型最终性能。

在模型设计上，本实验没有刻意追求复杂结构，而是采用较为基础的多层感知机作为预测模型。通过合理构造输入特征、规范处理训练折与验证折之间的关系，并在标签空间上采用对数变换，最终得到了较稳定的本地分数和较好的 Kaggle 线上成绩。这说明对于初学阶段的回归任务学习而言，先掌握完整而规范的建模流程，比单纯追求更复杂的模型更为重要。

本次实验的主要收获包括：理解了混合类型表格数据的基本处理方法；掌握了使用 `ColumnTransformer` 与 `Pipeline` 构建预处理流程的方法；理解了 K 折交叉验证、OOF 预测和多折平均提交的意义；掌握了使用 PyTorch 构建并训练多层感知机回归模型的基本步骤。

若进一步改进本实验，还可以从以下几个方面继续展开：一是对高基数类别尝试嵌入表示或目标编码；二是对简介文本引入更细致的统计特征或文本向量化方法；三是继续调整网络宽度、正则化系数与学习率等超参数，以进一步优化线上表现。