



**模式识别大作业**

题 目 房价预测

学 院 信息科学与工程学院

专 业 控制科学与工程

组 员 傅煦嘉

指导教师 赵海涛

**完成日期： 2019年12月12日**

**引 言**

模式识别(Pattern Recognition)，就是通过计算机用数学技术方法来研究模式的自动处理和判读。我们把环境与客体统称为“模式”。随着计算技术的发展人类有可能研究复杂的信息处理过程。信息处理过程的一个重要形式是生命体对环境及客体的识别。

通过模式识别的学习，我们对模式识别有了一定的了解和理解；模式识别属于人工智能的元老了，它需要人类主动去描述某些特征给机器。其应用领域也非常之广，主要在以下几个部分：语音识别与理解、文本—字符识别、生物特征识别、生物医学信号识别、图像检索、智能监控、军事跟踪与识别等。

我们本次作业运用模式识别知识根据房子的位置和大小等因素对房价进行预测。

**一、题目简介与任务**

**1.题目描述**

如果让购房者描述他们梦想中的房子，他们可能不会从地下室天花板的高度或靠近东西铁路开始。但这个游乐场竞赛的数据集证明，影响价格谈判的因素远远超过卧室数量或白色尖桩篱笆。79个特征变量描述(几乎)爱荷华州艾姆斯的住宅的每个方面。

**2.数据集**

本题目主要有训练集和测试集组成，每一个ID的房子都列出97个特征，但是有些数据集对应的一些特征为0；后面会对这些的数据做一些预处理。

**3.任务**

本次作业就是需要根据这79个特征来进行房价的预测。你的工作是预测每套房子的售价。对于测试集中的每个Id，您必须预测售出价格变量的值。

**二、整体解决方案**

**1.数据预处理**

这里数据清洗的工作有以下几个步骤：离群点，偏度校正，数据的缺失值处理，特征初步处理（删除，融合及投影）。

1.1.离群点

离群点：离群点是指一个序列中，远离序列的一般水平的极端大值和极端小值。它的产生是由于系统受外部干扰而造成的，会直接影响模型的拟合精度。GrLiveArea这一特征表示的含义是可居住的地上总面积。如图1是GrLiveArea取不同值时价格的分布情况，可以看出在大于4000部分有4个离群点，在这里我们并不了解数据集中这4个值是否准确，但是为了防止它对于最后精度的影响，选择将大于4000的点直接删除。删除后的结果如图2。

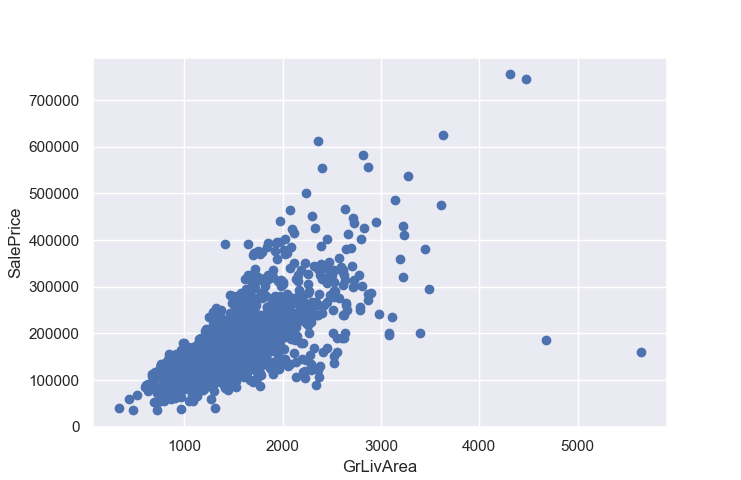
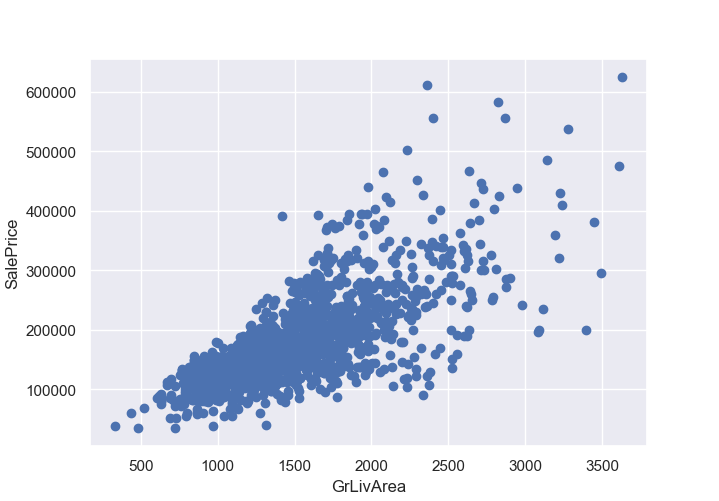
 

图1 存在离群点图 图2 删去离群点图

1.2.偏度校正

偏度，也称偏态系数，是统计数据分布偏斜方向和程度的度量，是统计数据分布非对称程度的数字特征，可以通过它判断数据分布的不对称方向以及程度，定义上偏度是样本的三阶标准化矩。

本训练集得出来的数据是正偏态分布，为了使数据的偏态系数变小使其更趋向于正态分布，我们所有偏态系数较大的用数字表示的特征进行对数变换处理（BOXCOX函数），转换后的模型满足线性性、独立性、方差齐性及正态性的同时，又不丢失信息。此处以偏度值0.5为标准，经过变换后的特征的偏度值都有了明显的改善。

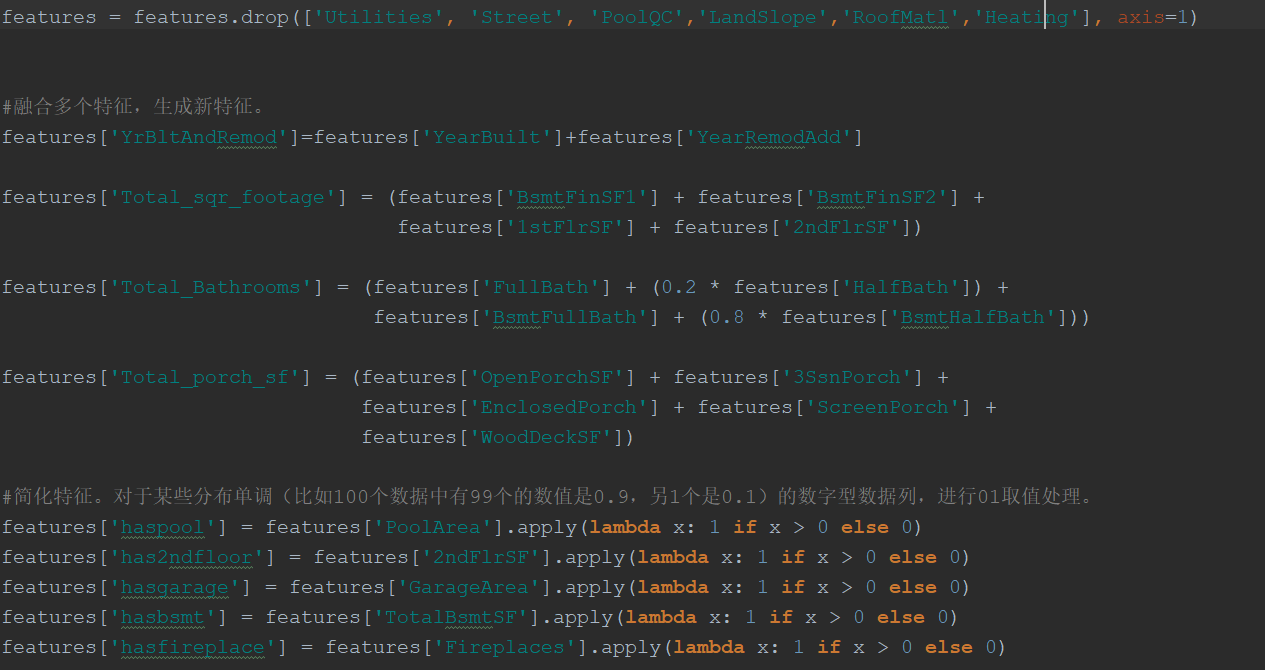
1.3. 缺失值

缺失值是指粗糙数据中由于缺少信息而造成的数据的聚类、分组、删失或截断。它指的是现有数据集中某个或某些属性的值是不完全的。缺失值的产生的原因多种多样，主要分为机械原因和人为原因。常用的插补缺失值方法有均值插补、极大似然估计、多重插补等。在均值插补中，将数据的属性分为定距型和非定距型。如果缺失值是定距型的，就以该属性存在值的平均值来插补缺失的值；如果缺失值是非定距型的，就根据统计学中的众数原理，用该属性的众数(即出现频率最高的值)来补齐缺失的值；而多重插补同均值插补的方法都属于单值插补，不同的是，它用层次聚类模型预测缺失变量的类型，再以该类型的均值插补。假设为信息完全的变量，Y为存在缺失值的变量，那么首先对X或其子集行聚类，然后按缺失个案所属类来插补不同类的均值。如果在以后统计分析中还需以引入的解释变量和Y做分析，那么这种插补方法将在模型中引入自相关，给分析造成障碍。

在此数据集中，在分布较为集中的特征中，我们对目标类的特征的缺失值补为None，对整数型或是浮点型的特征列我们用0补上缺失值；而分布较为分散的特征，如Exterior1st与Exterior2nd等，它们表示房屋外墙的材料，特征比较多且都为目标类，所以我们使用众数（即最普遍的）来填补缺值；对于LotFrongtage，由于取值较为分散，没有十分显著能代表其他特征的值，所以我们用平均值来补缺值。

1.4. 特征处理

本次实验的数据集中有79个特征，数量较多且质量参差不齐，部分特征中某一特征量过度集中，对于结果影响很小；部分特征相关度高，可以合并成为一个特征，合并出的特征会对结果的预测产生更好的效果；而还有一部分特征分布比较单调且为数字型序列，我们就把它做01处理。如果在对一些特征进行融合后产生新特征后，将旧特征去掉的话，准确率会有一点降低，运行速度也没显著变快，因此我们选择保存初始特征。



但是，有很大一部分特征是非数字量特征，我们需要将其转化成为数字才能被计算机识别。因此，我们使用了one-hot编码方式。One-Hot编码，又称为一位有效编码，主要是采用N位状态寄存器来对N个状态进行编码，每个状态都由他独立的寄存器位，并且在任意时候只有一位有效。它要求将分类值映射到整数值，之后每个整数值被表示为二进制向量，除了整数的索引之外，它都是零值，它被标记为1。在python中，我们使用get\_dummies函数对特征进行投影操作，得到处理后的最终特征。

**2.学习模型**

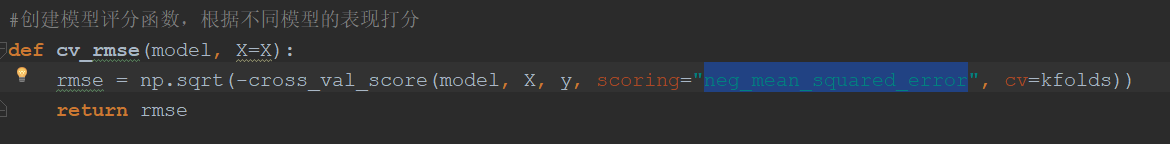
（1）交叉验证

交叉验证是用于防止模型过于复杂而引起过拟合，有时亦称为循环估计。我们先把数据样本切割成若干个较小子集，然后对一个子集进行分析，而其他子集则用来做后续对此分析的确认及验证。交叉验证法要求训练集的比例足够多，且训练集和测试集要均匀抽样。

交叉验证有很多种，我们本次采用了k折交叉验证，它的优点在于所有的样本都被作为了训练集和测试集，每个样本都被验证过一次。我们使用的python中sklearn包中Kfold函数，通常把一个样本分为10份；如果每次分类结果一样，那么最终产生的模型没有泛化能力，所以需要每次分类结果不同，即将参数Shuffle设置为True；但如果每次结果不同，很难重现之前的分类结果，故需要搭配使用random\_state随机种子数。

（2）评分函数

在定义好分类方法后，需要有一个标准来评定模型的准确率。我们使用cross\_val\_score来进行交叉验证，评分标准采用neg\_mean\_squared\_error。



（3）个体机器学习模型的创建

本次实验选用的个体机器学习模型有以下几个：  
【1】ridge（L2）岭回归模型

岭回归是为了解决线性回归中出现的过拟合以及在通过正规方程方法求Ɵ的过程中出现的不可逆两类问题而提出的，方法是引入正则化项，具体方程为：

其中为在损失函数上添加的一个惩罚项，称为L2正则化。当为0时，得到的是最小二乘解；当趋向更大时，岭回归系数就会趋于0。

【2】LASSO回归模型

LASSO模型与岭回归模型提出的原因相同，但相比较于岭回归，它采用的方法是L1正则化，具体方程为：

添加的是一阶的绝对值正则化项，它可以起到更好的筛选变量的作用。

【3】SVR支持向量回归模型

与支持向量机相反，SVR是希望训练集中的每个点都能尽量拟合到一个线性模型中，即在松弛变量内的为无损失的，而在松弛变量范围外的为有损失的。我们调用包中的SVR模块，其中C表示惩罚因子，C越大则margin越小，容易过拟合；而C过小则可能magin过大，容易欠拟合。gamma值必须大于0，并且随着增大容易繁华误差出现过拟合。

【4】boosting模型

Boosting的学习思路即，在给定一个训练数据集使，求一个弱学习器比求一个强学习器要容易得多，而boosting方法就是从弱学习器出发，反复学习，得到一系列弱学习器后，然后将弱学习器进行组合得到一个强学习器。本次实验采用了梯度上升、lightgbm与xgboost模型。

梯度上升是利用损失函数的负梯度在当前模型的值作为回归问题提升树算法中残差的近似值，拟合一个回归树。梯度上升本质上就是通过累加弱学习器来梯度下降，最终达到尽可能拟合掉误差。在python中调用函数，弱学习器的个数过小容易欠拟合，过大会过拟合且运行时间过长，经尝试设置为1000较为合适；学习率即每个弱学习器的权重缩减系数，较小的学习率意味着更多的迭代次数，这个参数需要与弱学习器的个数相匹配；损失函数部分我们使用huber，它的抗干扰能力较强，因为这个数据集并不完全准确，所以选择huber较为合适。

Lightgbm也称轻梯度上升，它相比较与梯度上升，拥有更快的训练速度，更低的内存消耗和更好的准确率。在实验中程序内加入时间戳，Lightgbm相比较于梯度上升在设置同样迭代次数的情况下，运行时间显著变短。在python中调用函数，与梯度上升类似，也将弱学习器数量设为1000，学习率设为0.01；提高精度最重要的参数是max\_depth和num\_leaves，num\_leaves影响整个调节树的复杂程度；feature\_fraction是用来防止过拟合及提高训练速度，bagging\_fraction+bagging\_freq同样也是用来降低过拟合的。

Xgboost最大的特点在于它能够自动利用CPU的多线程进行并行，同时在算法上加以改进和提高精度。

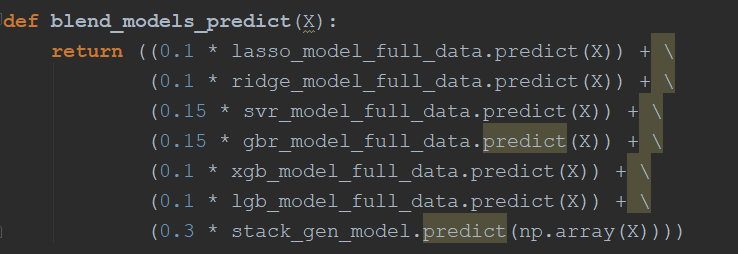
（4）集成学习

集成学习是通过一定的结合策略，将个体学习器综合起来最终得到一个强学习器，以达到博采众长的目的。集成学习可分为boosting、bagging和stacking。在stacking中，算法分为2层；第一层是不同算法形成的多个弱学习器，并产生与原始数据集等大的新数据集；在第二层用一个新的学习器去处理第一层的输出数据，最后得到结果。即将训练好的所有基模型对训练基进行预测，第j个基模型对第i个训练样本的预测值将作为新的训练集中第i个样本的第j个特征值，最后基于新的训练集进行训练。它的优点在于相比于独立的预测模型，它拥有更强的非线性表述能力，且能够降低泛化误差。

**3、融合个体学习器的结果并验证最后结果**

对每个模型进行交叉验证，并对不同模型的表现进行打分，由于是交叉验证，将使用不同的数据集对统一模型评分，因此每个模型对应一个得分序列，展示的为评分的均值与标准差。然后，使用训练数据作为输入，训练数据对数处理后的预测房价作为输出，进行各个模型的训练。

之后，根据各个模型的得分情况，配上相对应的权重，产生多模型组合学习器的预测值。



在训练确认完成所有模型后，我们将测试集的特征矩阵作为输入，传入训练好的模型，并进行价格的对数还原，插入到写入的csv文件的第二列，即价格列。

为了减小误差，对融合后的结果进行处理，即把极端值进行处理。我们使用pandas中的quantile函数，设置分位置为0.005和0.995，即位于前0.5%的价格，属于过大值，将其乘一个小于1的系数；而对位于后0.5%的数据，则属于过小值，将其乘一个大于1的系数。这样就完成了极端值的剔除，输出最终结果。

**三、作业总结**

对于这次作业我纠结了很久，不知道用哪个算法比较好，对python的包也不够了解如何使用。刚开始尝试做MINST手写数字，但是在pycharm总是容易崩溃，所以最后选择了房价预测这个数据处理预测的题目。

这次使用的模型并不多，且boosting的模型中设置的弱学习器的数量应该可以再进行优化；另外也没有尝试使用Adaboosting的模型，之后会尝试添加上看会不会有更好的效果。

这次的大作业让增加了我对模式识别的兴趣，对python编程有了初步的了解。同时感谢赵老师教学帮助，并且在作业过程中给了我们大家很多指导。