# 项目一：房价预测

## 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，偏态系数，数据的缺失值处理。

#### 1.1.离群点

离群点：离群点是由于系统受外部干扰而造成的。但是，形成离群点的系统外部干扰是多种多样的。首先可能是采样中的误差，如记录的偏误，工作人员出现笔误，计算错误等，都有可能产生极端大值或者极端小值。其次可能是被研究现象本身由于受各种偶然非正常的因素影响而引起的，例如。在人口死亡序列中，由于某年发生了地震，使该年度死亡人数剧增，形成离群点；在股

票价格序列中，由于受某项政策出台或某种谣传的刺激，都会出现极增，极减现象，变现为序列中的离群点。

这里离群点处理的办法是将 GrLiveArea 大于 4000 的点直接删除，如下图 1 为存在离群点图，图 2 为删去离群点图。

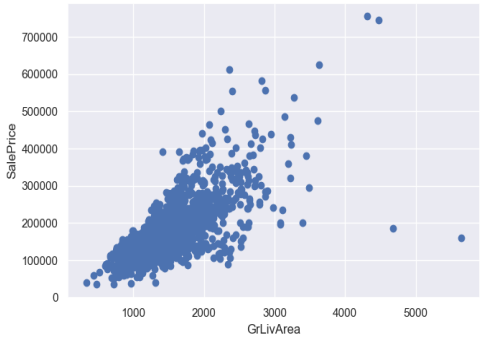


图 1 存在离群点图

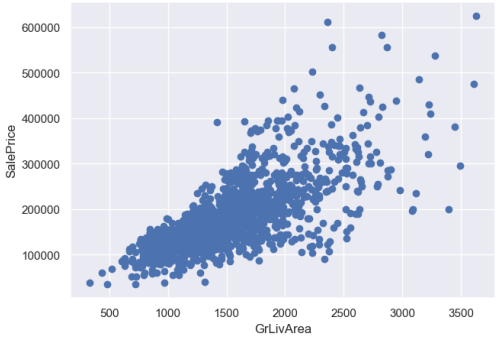


图 2 删去离群点图

## 1.2.偏态分布

偏态分布指频数分布的高峰位于一侧，尾部向另一侧延伸的分布。它分为正偏态和负偏态。偏态分布的资料有时取对数后可以转化为正态分布，反映偏态分布的集中趋势往往用中位数。

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，这里对其取对数函数，以使它的偏态系数变小。处理前的偏态图如图 3 所示，处理后的偏态图如图 4 所示；可以看出偏态系数由开始的 1.57 变为了 0.07，效果还是很明显的。

偏态图进行处理过后，确定特征与特征之间的关联系数，我们这里用热点图来将关联系数进行可视化；图 5 为特征之间的关联系数。

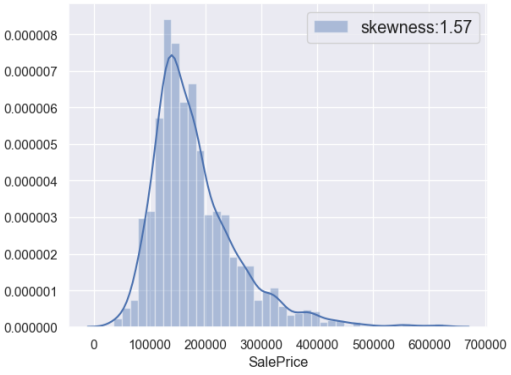


图 3 处理前的偏态图

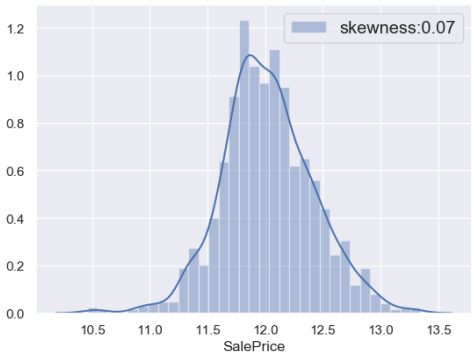


图 4 处理后的偏态图

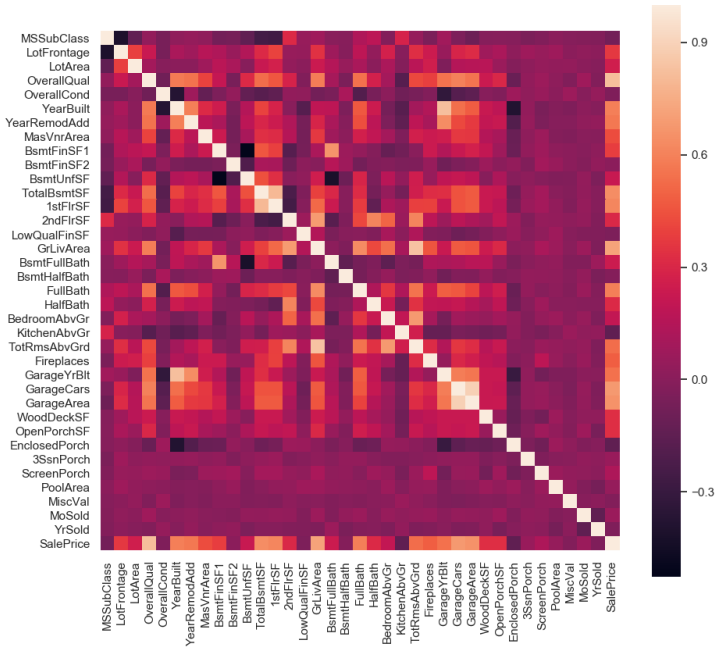


图 5 特征之间的关联系数

## 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。

机械原因是由于机械原因导致的数据收集或保存的失败造成的数据缺失，比如数据存储的失败，存储器损坏，机械故障导致某段时间数据未能收集（对于定时数据采集而言）。

人为原因是由于人的主观失误、历史局限或有意隐瞒造成的数据缺失，比如，在市场调查中被访人拒绝透露相关问题的答案，或者回答的问题是无效的，数据录入人员失误漏录了数据。

常用的插补缺失值方法：

(1)均值插补。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。

(2)利用同类均值插补。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 $X=(X_1,X_2,\cdots X_p)$ 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。

(3)极大似然估计（Max Likelihood ，ML）。在缺失类型为随机缺失的条件下，假设模型对于完整的样本是正确的，那么通过观测数据的边际分布可以对未知参数进行极大似然估计（Little and Rubin）。这种方法也被称为忽略缺失值的极大似然估计，对于极大似然的参数估计实际中常采用的计算方法是期望值最大化(Expectation Maximization，EM）。该方法比删除个案和单值插补更有吸引力，它一个重要前提：适用于大样本。有效样本的数量足够以保证 ML 估计值是渐近无偏的并服从正态分布。但是这种方法可能会陷入局部极值，收敛速度也不是很快，并且计算很复杂。

(4)多重插补（Multiple Imputation，MI）。多值插补的思想来源于贝叶斯估计，认为待插补的值是随机的，它的值来自于已观测到的值。具体实践上通常是估计出待插补的值，然后再加上不同的噪声，形成多组可选插补值。根据某种选择依据，选取最合适的插补值。多重插补方法分为三个步骤：①为每个空值产生一套可能的插补值，这些值反映了无响应模型的不确定性；每个值都可以被用来插补数据集中的缺失值，产生若干个完整数据集合。②每个插补数据集合都用针对完整数据集的统计方法进行统计分析。③对来自各个插补数据集的结果，根据评分函数进行选择，产生最终的插补值

在此数据集中，我们将有些特征数据的 null 值代表对应样例没有特定属性的称为缺失值原因一，其他原因造成的数据不完整称为原因二；为了数据处理比较方便，对于原因一造成的数据缺失我们用“U”或“0”和均值插补中的均值填充，原因二造成的数据缺失我们用均值插补的众数原理进行填充。

## 2.嵌入式选择与 L1 正则化

在过滤式和包裹式特征选择方法中，特征选择过程与学习训练过程有明显的分别；与此不同，嵌入式特征选择是将特征选择过程与学习器训练过程融为一体，两者在同一个优化过程中完成，即在学习器训练过程中自动地进行了特征选择。

给定数据集 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_m,y_m)\}$，其中 $x\in R^d$，$y\in R$。我们考虑最简单的线性回归模型，以平方误差为损失函数，则优化目标为

$$\min_{w}\sum_{i=1}^{m}(y_i - w^T x_i)^2 \tag{0-1}$$

当样本特征很多，而样本数相对较少时，式(0-1)很容易陷入过拟合。为了缓解过拟合问题，可对式(0-1)引入正则化项。若使用 L2 范数正则化，则有

$$\min_{w}\sum_{i=1}^{m}(y_i - w^T x_i)^2 + \lambda\|w\|_2^2 \tag{0-2}$$

其中正则化参数$\lambda$>0。式(0-2)称为“岭回归”，通过引入 L2 范数正则化，确能显著降低过拟合的风险。

那么，能否将正则化项中的 L2 范数替换为 Lp 范数呢？答案是肯定的。若令 p=1，即采用 L1 范数，则有

$$\min_{w}\sum_{i=1}^{m}(y_i - w^T x_i)^2 + \lambda\|w\|_1 \tag{0-3}$$

其中正则化参数$\lambda$>0。式(0-3)称为 LASSO(Least Absolute Shrinkage and Selection Operator)。

L1 范数和 L2 范数正则化都有助于降低过拟合风险，但前者还会带来一个额外的好处：它比后者更易于获得“稀疏”解，即它求得的$w$ 会有更少的非零分量。

为了理解这一点，我们来看一个直观的例子；假定$x$ 仅有两个属性，于是无论式(0-2)还是(0-3)解出的$w$ 都只有两个分量，即$w_1, w_2$，我们将其作为两个坐标轴，然后在图中绘制出(0-2)与(0-3)的第一项的“等值线”，即在($w_1, w_2$)空间中平方误差项取值相同的点的连线，如图 6666，式(0-2)与(0-3)的解要在平方误差项预正则化项之间折中，即出现在图中平方误差项等值线与正则化项等值线相交处。由图 2312 可看出，采用 $L_1$ 范数时平方误差项等值线与正则化项等值线的交点常出现在坐标轴上，即$w_1$或$w_2$为 0，而在采用 $L_2$ 范数时，两者的交点常出现在某个象限中，即$w_1$或$w_2$均非 0；换而言之，采用 $L_1$ 范数比 $L_2$ 范数更易于得到稀疏解。

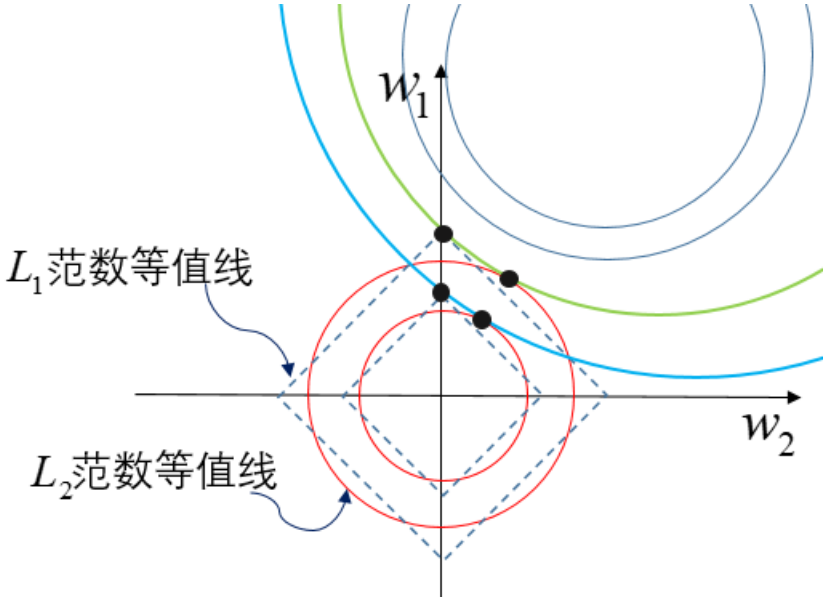


图 6 L1 正则化比 L2 正则化更易于得到稀疏解

注意到 $w$ 取得稀疏解意味着初始的 d 个特征中仅有对应着 $w$ 的非零分量的特征才会出现在最终模型中，于是，求解 $L_1$ 范数正则化的结果是得到了仅采用一部分初始特征的模型；换而言之，基于 $L_1$ 正则化的学习方法就是一种嵌入式特征选择方法，其特征选择过程与学习器训练过程融为一体，同时完成。

$L_1$ 正则化问题的求解可使用近端梯度下降(Proximal Gradient Descent,简称 DGP)。具体来说，令 $\nabla$ 表示微分算子，对优化目标

$$\min_x f(x)+\lambda\|x\|_1 \tag{0-4}$$

若 $f(x)$ 可导，且 $\nabla f$ 满足 L-Lipschitz 条件，即存在常数 L>0 使得

$$\|\nabla f(x')-\nabla f(x)\|_2^2 \le L\|x'-x\|_2^2 \ (\forall x, x') \tag{0-5}$$

则在 $x_k$ 附近可将 $f(x)$ 通过二阶泰勒展开式近似为

$$\hat{f}(x) \approx f(x_k)+\langle \nabla f(x_k), x-x_k\rangle+\frac{L}{2}\|x-x_k\|^2=\frac{L}{2}\left\|x-\left(x_k-\frac{1}{L}\nabla f(x_k)\right)\right\|_2^2+const \tag{0-6}$$

其中 const 是与 $x$ 无关的常数，$\langle\cdot,\cdot\rangle$ 表示内积。显然，式(0-6)的最小值在如下 $x_{k+1}$ 获得：

$$x_{k+1}=x_k-\frac{1}{L}\nabla f(x_k) \tag{0-7}$$

于是，若通过梯度下降对 $f(x)$ 进行最小化，则每一步梯度下降迭代实际上等价于最小二次函数 $\hat{f}(x)$。将这个思想推广到式(0-4)，则能类似地得到其每一步迭代应为

$$x_{k+1}=\arg\min_x \frac{L}{2}\left\|x-\left(x_k-\frac{1}{L}\nabla f(x_k)\right)\right\|_2^2+\lambda\|x\|_1 \tag{0-8}$$

即在每一步对 $f(x)$ 进行梯度下降迭代的同时考虑 $L_1$ 范数最小化。

对于式(0-8)，可先计算 $z = x_k - \frac{1}{L}\nabla f(x_k)$，然后求解

$$x_{k+1} = \arg\min_x \frac{L}{2}\|x - z\|_2^2 + \lambda\|x\|_1 \tag{0-9}$$

令 $x^i$ 表示 $x$ 的第 $i$ 个分量，将式(0-9)按分量展开可看出，其中不存在 $x^i x^j$ $(i \neq j)$这样的项，即 $x$ 的各分量互不影响，于是式(0-9)有闭式解

$$x_{k+1}^i = \begin{cases} z^i - \lambda / L, \lambda / L < z^i; \\ 0, |z^i| \leq \lambda / L; \\ z^i + \lambda / L, z^i < -\lambda / L \end{cases} \tag{0-10}$$

其中 $x_{k+1}^i$ 与 $z^i$ 分别是 $x_{k+1}$ 与 $z$ 的第 $i$ 个分量。因此，通过 PCG 能使 LSSAO 和其他基于 $L_1$ 范数最小化的方法得以快速求解。

# 3.交叉验证

如果给定的样本数据充分，进行模型选择的一种简单方法是随机地将数据集切分成三部分，分别为训练集(training set)、验证集(validation set)和测试集(test set)。训练集用来训练模型，验证集用来模型的选择，而测试集用于最终对学习方法的评估。在学习到不同复杂度的模型中，选择对验证集有最小预测误差的模型。由于验证集有足够多的数据，用它对模型进行选择也是有效的。

但是，在许多实际应用中数据是不充足的，为了选择好的模型，可以采用交叉验证方法。交叉验证的基本想法是重复地使用数据；把给定的数据进行切分，将切分的数据集组合为训练集与测试集，在此基础上反复地进行训练、测试以及模型选择。

3.1 简单交叉验证

简单交叉验证方法是：首先随机地将已给数据分为两部分，一部分作为训练集，另一部分作为测试集(例如，70%的数据为训练集，30%的数据为测试集)；然后用训练集在各种条件下(例如，不同的参数个数)训练模型，从而得到不同的模型；在测试集上评价各个模型的测试误差，选出测试误差最小的模型。

3.2 S 折交叉验证

应用最多的是 S 折交叉验证，方法如下：首先随机地将已给数据切分为 S 个互不相交地大小相同的子集；然后利用 S-1 个子集的数据训练模型，利用余下的子集测试模型；将这一过程对可能的 S 种选择重复进行；最后选出 S 次评测中平均测试误差最小的模型。

3.3 留一交叉验证

S 折交叉验证的特殊情形是 S=N，称为留一交叉验证，往往在数据缺乏的情况下使用。这里，N 是给定数据集的容量。

本次实验用的是 S 折交叉验证，我们将 S 取为 5 进行交叉验证，同时调用的是 sklearn 包，直接将参数中的 cv 设为 5。

# 三、实验结果

实验结果如图 7 所示，可以看出 testing score 的误差带宽度比较大，而 training score 的误差带相比小很多，但随着数据量的增大，图形呈收敛趋势，同时存在欠拟合状态。

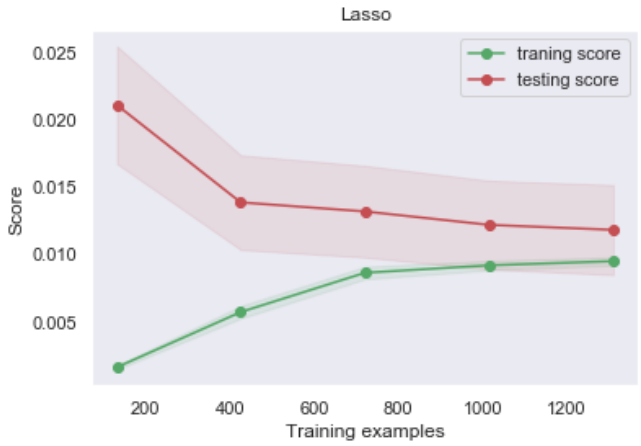


图 7 实验结果