数据挖掘期末大作业报告：艾奥瓦房价预测

窦鹏飞 1500017707 李泽坤 1500017841 伍维晨 1500017805 严祚宇 1500017844

(按音序排名)

一、数据挖掘任务

本次大作业中，本组使用 kaggle 竞赛上的艾奥瓦州房价预测数据集(https://www.kaggle.com/c/iowa-house-prices-regression-techniques)，利用房屋的各种属性和市场情况，对房屋价格进行预测。数据挖掘的任务有两方面：其一是发现与房屋价格紧密相关的各类因素，其二是构建回归模型以尽量精确地预测房价。

二、数据集描述

数据集包含1460条记录，每条记录对应一套房产，包含其价格和其他77个属性，记录了有关房屋的各种信息。

观察数据集，可以发现这77个属性中包含34个定比变量（例如地上居住面积、地下室面积、车库停车位数量），22个定序变量（如房屋外层结构的维持状态、地下室状态）和21个定类变量（如房屋所属地段的类别）。所有变量的含义描述，可参见data_description.txt文档。对于不同类型的变量，需要进行不同方式的预处理。我们用数据集中的前1160条记录构建模型，用后300条记录检测模型的效果。

三、数据预处理

1.缺失值处理

数据集中绝大多数的定比变量都不存在缺失值。对于少数定比变量存在的少数缺失值，我们简单地以均值代替之。

在定序变量方面，数据集中有几个定序变量涉及对房屋各方面的评价，例如对地下室状态的评价、对车库状态的评价、对游泳池状态的评价等等。但是，对于没有地下室、或没有车库、或没有游泳池的房产来说，这些属性自然是缺失的。在这种情况下，我们直接把缺失值视作最差的一种评价即可。

一个比较特殊的情况是，"GarageYrBlt"属性代表车库建造的年份，没有车库的房屋在这一属性上自然缺失。从直觉上讲，车库建造的年份与房屋价格之间也并没有太紧密的关联，有关车库的信息在其他一些变量，例如车库面积（无车库则为0）、车库停车位数量（无车库则为0）、车库状态（无车库视为最差）中得到了充分体现，于是我们决定直接删除"GarageYrBlt"这一属性。

在定类变量方面，有些属性值显示为"NA"，但其实并不代表数据缺失。例如，"Alley"变量代表“连接到房屋的后街的类型”，若房屋不与后街相连则显示为"NA"。对此，我们可以直接把这种情况也视作一个普通的类别。也有些显示为"NA"的定类属性确实代表数据的缺失，对此在无从查找真实值的情况下，我们也只能将其视为一“类”进行处理。

2.定序变量与定类变量的数值化

（1）对于所有定序变量，我们用简单的打分方式进行数值化处理，例如将“非常好”设为5分，“好”设为4分，“平均”设为3分，“一般”设为2分，“差”设为1分，“没有”设为0分等；

（2）一部分定类变量之间实际上存在着质量上的序关系。例如，“车库类别”这一变量分为无停车位、外设停车位、屋外地上车库、屋外地下车库、与房屋相连接的车库、房屋地下车库、两种以上车库几类。从直觉上看，这几个类别对应的车库档次是依次上升的，因此我们对这一变量也采取了打分的方式进行数值化；

（3）对于实际上也没有序关系的纯粹的定类变量，采用one-hot编码的方式进行数值化；

3.新变量的构建

考察各变量的含义后，我们用以下几种方式构建新变量：

（1）对各类评价得分进行加权平均或交互相乘，得到综合评价。例如，停车位数量、车库种类、车库建造完成情况三个变量都描述的是车库的状况，我们把这三个变量加权平均，得到了对车库的总体评价；又如，对房屋外层结构、地下室、厨房和供暖系统的评价得分取平均，得到对房屋的总体评价；再如，将壁炉质量分数与壁炉数量相乘，得到“壁炉总得分 ” 等。

（2）加总相关的变量，例如将地上、地下的房间数相加，得到房间总数等；

（3）生成其他感兴趣的变量，例如用交易年份减去房屋建成年份得到房龄，用总居住面积除以房间数得到平均房间面积等；

4.孤立点分析

根据经验，房屋最主要的功能是居住，居住面积是决定房屋价格的最重要因素之一。我们日常所谈论的“房价”，一般指的是单位居住面积的价格。为此，我们用销售价格除以居住面积，得到单位居住面积的平均价格。对这一属性进行分析，可以发现有5套房产的单位居住面积价格低于40，显著地小于其他房产，我们因此决定删去这5条记录。

5.数据归一化

我们对所有数值变量进行了z-score规范化处理，即取

$$z_i = \frac{x_i - \mu}{\sigma}$$

其中，x_i为规范化前的属性值，z_i为规范化后的属性值，μ为样本均值，σ为样本标准差。

四、线性回归分析(OLS)

在这一部分中，我们用最简单的线性回归(Ordinary Least Square,OLS)框架对数据进行分析，其目的是寻找出与房价在统计意义上显著相关的变量。回归模型为：

$$y = X\beta + \varepsilon$$

其中，y是被解释变量（房价），X是解释变量，β是回归系数，ε是随机误差。用最小二乘法进行参数估计：

$$\hat{\beta}_{OLS} = \mathbf{argmin}_{\beta} ||y - X\beta||_2^2$$

在预处理结束后，数据集中共有185个解释变量，去除共线性后，其中40个变量与房价存在着显著的正相关关系，6个变量与房价存在着显著的负相关关系（在0.05水平下）。限于篇幅，我们选取其中几个：

（1）OverallQual：表示房屋的“总体质量”，与房价显著正相关；这一点非常容易理解，也反映出评价体系的重要性；

（2）PoolArea:表示游泳池面积，与房价显著正相关；这一方面是由于游泳池本身的占地和建造成本，另一方面则是由于含有游泳池的房产一般都是所谓“豪宅”，价格不菲；

（3）kitchen_quality:表示厨房的质量，与房价显著正相关；厨房在很大程度上决定了生活的质量，厨房质量与房价显著正相关并不令人意外；

（4）fireplace_ttlscore:表示对壁炉的总体评价得分；

（5）Neighborhood_Crawfor,Neighborhood_NridgHt,Neighborhood_StoneBr：三个哑变量，表示房屋是否位于Crawfor,NridgeHt,StoneBr三个街区；这三个变量与房价正相关，表示这三个街区的房价整体显著较高；

（6）SaleType_New：哑变量，表示房屋在被销售时是不是新建的；该变量与房价显著正相关，表示新房的价格整体显著较高；

（7）age：表示销售时的房龄，与房价显著负相关，表明越老的房子越难卖出好价钱；

（8）LandSlope_Sev：哑变量，表示房屋所在的土地是否存在严重的倾斜，该变量与房价显著负相关。

OLS框架的完整结果，可参阅OLS results.txt文档。

五、模型选择

这一部分的目的是构建回归模型以期尽可能精确地预测出测试集中的房价。对模型进行评价的标准是其在测试集上预测结果的根均方误差(Root Mean Square Error,RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

上式中，N为测试集样本数量，$y_i(i=1,2,\ldots,N)$为每套房产的真实价格，$\hat{y}_i(i=1,2,\ldots,N)$为相应房价的预测值。很显然，$RMSE$越低的模型，预测效果越好。

在上一部分中，我们用普通线性回归模型和最小二乘法来估计回归参数、构建回归模型。这一模型虽能揭示出统计意义上与响应变量（房价）显著相关的解释变量，但是其预测效果并不好（$RMSE > 40000$）。因此，我们需要更有效的回归模型和参数估计方法。本组在这次作业中进行了以下尝试：

（一）线性回归模型的正则化方法（Lasso,Ridge,Elastic Net）

对于线性回归模型

$$y = X\beta + \varepsilon$$

可以用不同的方式进行参数估计，以实现变量选择或者减小估计量方差的目的。其中，使用最广泛的是正则化(regularization)方法，即在损失函数中加入对参数范数的惩罚项，例如：

Lasso:

$$\hat{\beta}_{Lasso} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n}||y - X\beta||_2^2 + \alpha||\beta||_1$$

Ridge Regression:

$$\hat{\beta}_{Ridge} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n}||y - X\beta||_2^2 + \alpha||\beta||_2^2$$

Elastic Net:

$$\hat{\beta}_{EN} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n}||y - X\beta||_2^2 + \alpha_1||\beta||_1 + \alpha_2||\beta||_2^2$$

以上三式中，n为训练集样本数目。通过调整正则化参数α，可以得到不同的参数估计以期降低预测误差。本组实验得到的最优结果如下：

参数估计方法	**最小RMSE**	**对应参数**
Lasso	**28045**	$\alpha = 130$
Ridge Regression	**29429**	$\alpha = 0.05$
Elastic Net	**29644**	$\alpha_1 = 0.12, \alpha_2 = 0.08$

（二）核岭回归(Kernel Ridge Regression,KRR)

核岭回归方法是指先通过核变换将解释变量X映射到高维空间中，以$\phi(X)$代之；然后再在高维空间中，以类似于岭回归的方法进行线性回归。其参数估计方法大体可表示为：

$$\hat{\beta}_{KRR} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n} ||y - \phi(X)\beta||_2^2 + \alpha||\beta||_2^2$$

通过调整核函数和正则化参数，可以降低模型的预测误差。本组实践得到的最优结果是，将核函数取为三次多项式，并取$\alpha = 4$，可以使模型的$RMSE$下降到22295.相对于简单线性模型的各种正则化方法，这是一个巨大的改进。

（三）回归树模型及其集成（Gradient Boosting Regression,GBR and eXtreme Gradient Boosting, XGB）

回归树模型(Regression Tree)是指用决策树的方法对数据进行回归分析。例如，一个2层的二叉决策树可以将所有样本分为$2^2 = 4$类，然后对每类样本赋予一个预测值。但是，当样本量和属性数都很大时，简单的回归树模型难以得到好的结果，因此需要进行集成(Boosting)。梯度集成法（Gradient Boosting）就是这样的一种集成方法，它通过迭代的方法训练出许多个回归树模型，然后用加权投票的方式给出一个最终预测。迭代开始前，所有训练样本被赋予相同的权重；每一轮迭代根据现有样本的权重训练出一个加权误差最小的回归树模型，同时得到该模型在最终结果中的投票权重（训练误差越大的模型，投票权重越小）；再根据本轮迭代中各样本的训练误差，更新各样本的权重，使训练误差越大的样本权重也越大，以期在此后的迭代中被“重点关注”。为了避免过拟合，在训练过程中可以采用采样法，即每轮迭代只随机选取一定比例的样本进行训练。通过调整模型的迭代次数（也就是回归树个数）、学习速率、回归树的树深和分叉数、采样比例等来降低测试误差。在本组的实验中，选取70%的采样比例、0.1为学习速率、用两层的二叉回归树进行440次迭代，最优情况下可以使测试误差降低到$RMSE = 19503$，但有较大的随机性。这相对于核岭回归模型，又是一个很大的提升。

XGB（eXtreme Gradent Boosting）方法可以说是提升方法的完全加强版本，在各大比赛中也展示了强大的威力。传统的Gradient Boosting在优化时只用到了一阶导数信息，XGB则对代价函数进行了二阶泰勒展开，同时用到了一阶和二阶导数，这使得迭代更有效率。同时，XGB在代价函数里加入了正则项，用于控制模型的复杂度。正则项里包含了树的叶子节点个数、每个叶子节点上输出的score的ℓ_2模的平方和。正则项降低了模型的随机成度，使学习出来的模型更加简单，也更加稳定，防止过拟合，这也是XGB优于传统Gradient Boosting的一个特性。XGB还带有shrinkage参数，它类似于学习速率，在每次对树的boosting中增加了一个权重，通过这种方式来减少每棵树的影响力，给后面的树提供空间去优化模型。同时树的个数对XGB优化结果基本没有影响，所以在本组实验中将树个数设为50以加快迭代速度，迭代轮次设为530防止过拟合，最终XGB测试误差为$RMSE = 19892$

（四）神经网络

人工神经网络是由大量处理单元互联组成的非线性、自适应信息处理系统。处理单元与人的神经元类似，接受多个输入，通过包含的传递函数产生一个输出。若干个处理单元构成一个网络层，若干个网络层构成一个神经网络。我们使用了Shallow结构的神经网络，这是一种前向神经网络，第一层为输入层（input layer），最后一层为输出层（output layer），中间有若干层隐藏层（hidden layer）。一个隐藏层由k个神经元构成，对应一个非线性函数变换：

$$f(x) = \sum_{i=1}^{k} c_i \sigma(a_i^T x + b_i) + c_0$$

神经元有两种状态：激活或未激活。在实际的神经网络中，一般使用激活函数$h(x)$来表示处理单元是否处于激活状态，我们选用relu函数作为激活函数:

$$h(x) = max(0, x)$$

在本组实验中，神经网络的表现并不好，$RMSE = 41634$ 。是因为特征过多，而数据和特征相比却相对稀少，所以出现了过拟合的问题。

（五）模型堆积(stacking)

在以上的部分中，本组尝试了多种回归模型。这些模型的估计性质很不相同，效果也各有千秋。有些模型会高估房价、另一些模型则会低估房价，如果能够结合不同模型的优点，“取长补短”，就能得到更好的预测效果。这就是模型堆积(stacking)所要完成的任务。

stacking步骤：

1. 将初始数据输入该算法，总共输入三个参数：初始模型构成的集合、train_data、test _data
2. 将train_data划分为k个fold，test_data不变
3. 对每个fold使用初始模型训练出一个新模型
4. 以k为5为例，利用其中4个fold训练出的模型分别对剩下一个fold的train_data进行预测，连接这5个预测值形成新训练集
5. 将之前训练出的5个模型分别对test_data进行预测并取均值，作为新的测试集
6. 将新的初始模型、新训练集（来自4）、新测试集（来自5）输入算法，重复步骤1到6，直到初始模型集合中没有模型，输出最终的对于test data的预测

stacking得到的最终结果：**18393**

结论：在本组数据中，结合不同模型可以得到比单模型更好的解。

六、总结与收获

总结

总的来说，我们的模型最终对于房价有了相当好的预测。通过对比预测值和真实值，我们惊喜地发现不仅数据的误差小了很多，而且更重要的是其序关系基本上得以保持，即无论是在低端房屋还是高端房屋，我们的模型预测能力都较强，且在保守与激进之间维持了很好的平衡。比如测试集中最贵的房屋售价高达72万，而我们的模型直接预测出了69万，这也是预测值中最高的。

同时，我们不仅预测得好，可解释性也得到了相当大的保留，这对于模型的最终可理解性是非常重要的。如对于车库和地下室的处理让我们看到了一些原始数据中没有的特征，这些属性对于房屋的售价还是起到了重要的影响作用的。

在潜在有用性方面，虽然美国地区的房屋数据和中国可能差异很大，但是预测的方法依然是可以通用的，只要有足够的数据，我们的模型构建和预处理过程依然可以用于中国房屋价格的预测，并给市场一定的参考作用。

收获

本次作业中，我们充分体验到了数据预处理的重要性，在数据科学领域，有一句很经典的话叫“Garbage in, Garbage out"。即使数据本身对目标有较好的解释效果，如果选取了不当的预处理策略或者没有投入足够的时间和精力去耐心”挖掘“，那么得到的模型上限依然不会太高。

正如老师所言，我们90%的时间都花在了数据预处理上。一开始我们直接采用最基本的预处理，不断尝试各种模型都效果不佳。后来我们根据OLS提供的指导，利用数据初步探索和相关系数等工具对数据集的每一列都展开了繁复详细的预处理探索，在此过程中和模型选择相结合，最终一点点取得进步，达到了较好的单一模型预测结果。

即便如此，单一模型依然无法取得质的飞跃，最终通过查阅文献等方式，我们采取了模型提升的方法来融合多种较为不错的模型。在综合了各模型的特点之后，得到了最好的结果。

通过本次作业，我们对于数据挖掘任务和流程有了更深刻的理解。数据挖掘远远不是尝试模型、调整参数那么简单。在朝着目标前进的过程中，最重要的还是从数据本身着手，用课程中讲到的挖掘步骤，脚踏实地地前进，在对模型和数据集有了更深的领悟后，我们也就实现了一开始的目标。