目录

1 解释分析数据

1.1 数据描述

我们的项目调查了来自kaggle.com的纽约市Airbnb公开数据。该数据集提供了在Airbnb上列出的属性的详细特征，如房价、社区、房间类型、是否空置和预订所需的最少天数。它还包括额外的信息，如房源描述和以前预订的评论。在预处理前，数据包含13个特征和48895个观测值。有了这个数据集，我们将研究不同的特征会如何影响一个房源的租价。经过适当的数据预处理，我们将尝试通过建立不同的模型(线性模型、多项式模型等)来捕捉这些关系。

1.2 数据可视化

1.2.1 变量的相关性

对于数值类型的特征，我们绘制了一个特征相关图来可视化房源的哪些特征在决定价格(单位:美元/夜)时相对重要。

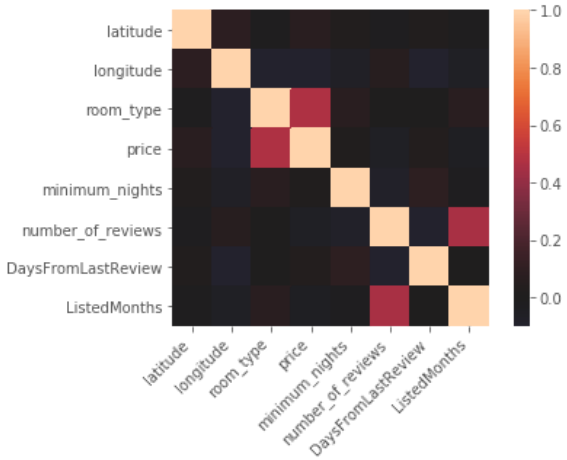

图1:特征相关图

如图1所示，价格与纬度和房间类型以及评论数量(过去的评论数量)高度相关。这种情况是符合实际的，因为曼哈顿的房源租金通常比其他县的房源租金要贵，一套完整的房源通常比一间共享房或一间包房要贵。此外，一些数字特征似乎与租金价格几乎没有相关性，因此在训练模型时使用正则化（套索回归）或变量选择是合适的。

1.2.2 Airbnb的价格

图2显示了Airbnb列出的房价分布。根据这个图，发现列出的价格分布是右倾的，我们通过对房价取log来修正数据的右倾。(我们将在后面的1.3.6部分讨论这种数据转换的原因)。

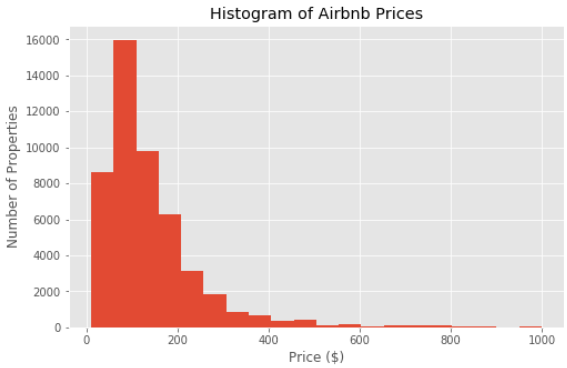

图2:Airbnb价格分布图

由于租金与纬度相关，我们绘制了纽约市周边五个邻近社区(曼哈顿、布朗克斯、皇后区、布鲁克林、斯塔顿岛)的租金价格分布。从图3可以看出，曼哈顿的平均房价最高，而布朗克斯区的平均房价最低。此外，与其他四个街区相比，曼哈顿的房价变化幅度最大。这些观察结果表明，包含房源的地理信息的特征可能有助于预测房源的租金。

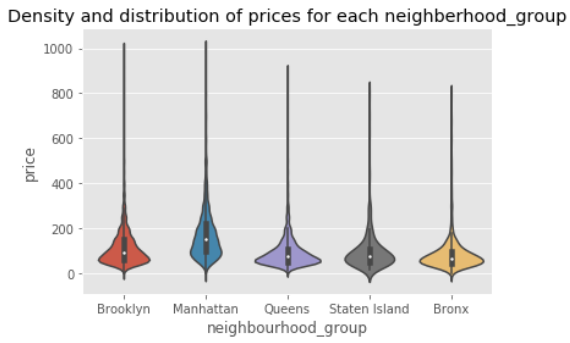

图3:各个县airbnb的价格

1.3 数据预处理

1.3.1 离群值

离群值的传统定义是指低于第一个四分位数或高于第三个四分位数的四分位数范围的1.5倍以上的数据点。然而，在airb2b房屋租金预测项目中，由于数据的性质，我们不采取这种方法。相反，我们只考虑基于常识的价格区间[10,1000]。(低于10美元的价格是不太可能的，而且几乎没有利润，1000美元以上的豪华房源也很少见。)结果，我们在48895个观测数据中剔除了250个异常值(数据的0.5%)。

1.3.2 顺序值

为了方便我们的回归建模，我们将“房间类型”列转换为序数值。这三种房间类型的平均价格分别是共享房间66美元、私人房间85美元和整套公寓193美元。这显示了房间类型之间的顺序关系，因此我们按升序价格为每种类型分别标记了1、2和3。

1.3.3 类别值

“街区组”和“街区”列包含类别值。由于“街区”指定了每个“街区组”中的区域，所以这两个列包含重叠的信息。为了在保持地理信息完整的情况下避免相关的问题，我们将这两列合并成一个新列(称为“街区”)。此外，由于不同街区值之间没有明显的序数关系，我们在新列上应用了one-hot编码，以方便我们的模型构建和预测。

1.3.4 附加特征

“最近一次评价时间”一栏隐藏着房源新鲜程度的信息，但是它的时间戳类型数值即不够直观，又不符合机器学习模型输入数值的要求。因此，我们通过计算当前日期与原始时间戳之间的差值来重塑这一列特征，该差值就是“距离上次评价的天数”。新列(“距离上次评价的天数”)给出了一个简单的、对读者友好的整数值，以度量最近的房屋评价情况。此外，我们还将“每月评价数量”替换为“月份数”(即“累计评价的数量”/“每月评价数量”)。这一新列数字显示了房源上市时间，这个新特征也与房价息息相关。

1.3.5 空缺值

由于数据的高度完整性，空缺值只出现在1.3.4的新列“距离上次评价的天数”中，这些新列继承了原来的“最近一次评价时间”和“每月评价数量”，很难确定这些新列中的缺失值是由于记录错误造成的，还是仅仅因为没有用户评价(我们认为这两个原因都是合理的)，所以我们用总体的中位数替换了这些缺失的值。由于我们的数据有一个明显的右偏的模式，平均替代在这种情况下是不合适的。对于新列“月份数”，“每月评价数量”中缺失值通常会导致新列出现缺失值，因此我们用0作为填充值。

1.3.6 响应变量转换

根据我们价格范围的直方图(介于$10和$1000之间)，响应变量是高度右偏的，大多数数值位于$10和$250之间，长尾向右。这与我们的常识相符，即多数人选择价格合理的房源，而不是高级奢华的房源。为了减少不正常的昂贵房源的过度影响预测结果，并稳定数据中的方差，我们可以通过对价格的值取对数来转换价格的分布。这项技巧将帮助我们在后期提高回归模型的准确性。

2 回归建模

2.1回归建模模型评价

在模型评价中，我们选择二次损失函数而不是其他损失函数。由于我们的响应变量（租

房价格）在数值上是连续的，在价格高估和价格低估之间有相对相等的损失(租房者和房东都要参考预测结果)。在这种情况下，二次损失在数学上更易于处理（容易求导），也更适合这种对称性质的方差。
具体来说，我们使用了预测价格对数和实际价格对数之间的均方根误差(RMSE)作为模型的评估指标。如前一节所述，通过对价格取对数，我们可以确保预测高价格和预测低价格的误差对结果的影响是一样的。

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

2.2 线性模型

我们首先通过求解最小二乘问题来拟合一个简单的线性模型，并将其作为我们项目的基线。在这种情况下，由于特征之间的线性关系和特征空间是不可逆的，所以该问题不存在唯一解。因此，我们需要加入一个正则化项来保证我们的线性模型的有效性。

2.2.1套索回归

首先，我们在最小二乘优化问题中加入一个l1正则项来保证线性模型有唯一解。将lasso模型作为我们的基本模型，除了可以保证线性模型有唯一解之外，也可以帮我们删除冗余变量，只保留与响应变量最相关的特征变量。

$$minimize \sum_{i=1}^{n}(y_i - w^\intercal x_i)^2 + \lambda \sum_{i=1}^{n} w \qquad (2)$$

然后,我们实现了一个5倍交叉验证,找到最好的λ参数，实现RMSE最小化。最优λ是0.0000045，测试集的RMSE是0.44915，训练集的RMSE是0.43924。我们可以看到训练误差和测试误差都很大，所以Lasso回归模型存在欠拟合的问题，我们的训练数据的特征不应该使用lasso进行变量选择。

2.2.2岭回归

为了防止欠拟合，我们用一个二次正则项代替了l1正则化项，这样我们既能保证唯一解，又能保留所有特征。

$$minimize \sum_{i=1}^{n}(y_i - w^\intercal x_i)^2 + \lambda \sum_{i=1}^{n} w^2 \qquad (3)$$

我们还实施了5倍交叉验证,去寻找最好的λ参数，实现RMSE最小化。如图4所示,最优参数λ是0.0000045，测试集的RMSE是0.44911。我们可以看到,岭回归模型和索回归模型相比，在测试数据的得分上有微小的进步。

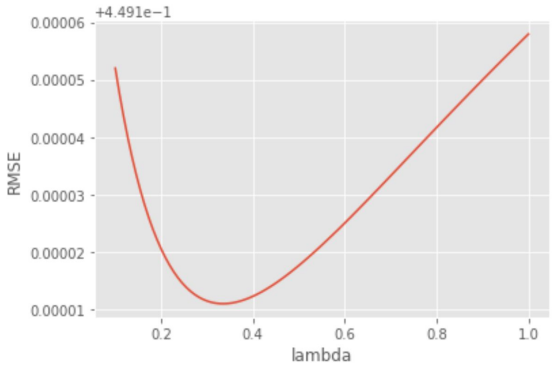

图4: 脊回归的5倍交叉验证

一般来说，使用线性模型捕捉价格与数据特征之间的关系，会产生相对较高的训练误差和测试误差，因此我们需要在数据集上找到更多的属性特征或使用更复杂的回归模型。因为我们的目的是为Airbnb的房源定租价，所以很自然会首先寻找其他房源特征数据，比如便利设施、附近的设施和过去的评论。然而，由于我们只有房源id，无法找到匹配的便利设施数据。由于在我们的项目中发现新的数据和特性比较困难，所以我们下一步采取了更复杂的回归模型。

2.3树模型

基于以前的结果，我们的建模仍然有改进的空间，因为在输入特征和响应之间可能存在

一些线性模型没有捕捉到的非线性关系。为了考虑这种可能性并获得更高的预测精度，我们将尝试更复杂的树的模型。由于考虑到数据的大小，单个树很可能不适合，因此我们决定使用bagging并增强树模型，以提高灵活性并控制方差。

2.3.1随机森林回归量

随机森林算法引入了决策树的套袋技术。本质上，它收集了大量独立的决策树，这些决策树作为一个整体运行，并通过分割随机的特征子集来将这些树去关联。这个算法的关键参数是树的数量，我们选择了10个不同的值来测试我们的回归变量:100、200、300、500、1000、1500、2000、3000、4000、5000。根据图5，通过选择2000种不同的树，我们可以实现的最低RMSE是0.48710。不幸的是，在这种情况下，随机森林的测试得分并没有优于的正则化线性模型(事实上，比线性模型的性能更差)。这部分是由于我们的数据集缺乏有效特性。随机森林模型是一种更加灵活的模型，它通常适用于具有足够数量特征的大数据集，而我们现有数据集的有限数量的特征可能无法充分体现该算法的强大功能。

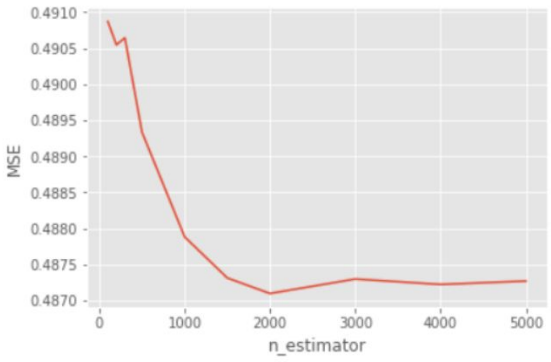

图5:具有不同树数的随机森林的RMSEs

2.3.2梯度增强回归模型

随机森林模型对低偏差、高方差的大树取平均，而增强技术则将高偏差、低方差的小树结合起来。boosting的树模型是使用前一棵树的信息，逐个生长的。其中每一棵回归树学习的是之前所有树的结论和残差，拟合得到一个当前的残差回归树。其中残差=真实值-预测值，提升树即是整个迭代过程生成的回归树的累加，最终实现较低的总体偏差。第一次尝试中，我们试图拟合一个常规的梯度增强回归模型。传统梯度增强的一个潜在问题是，它可能学习得太快，从而导致对数据的过度拟合。为了解决这个问题，我们调整了许多不同的增强参数，并保持较低的学习率(10个值严格地在0.01和0.1之间)。通过对测试数据集的拟合，我们发现最优学习率为0.07，对应的交叉验证误差为0.40907。这误差明显低于之前的线性模型，因此在数据集上使用梯度增强模型确实有助于解决欠拟合问题。此外，与采用套袋技术的随机森林回归模型相比，梯度增强回归模型具有更大的灵活性，在预测精度上也有显著提高。

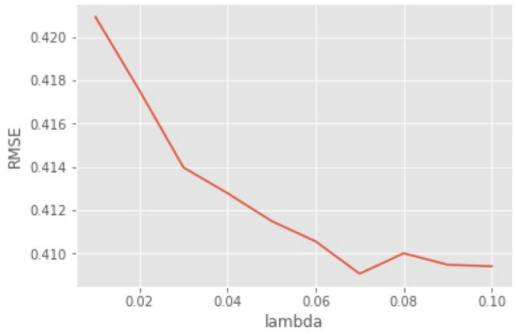

图6:不同学习率的梯度增强回归模型的RMSEs

2.3.3 XGBoost回归量

XGBoost (Extreme Gradient boost)是梯度增强算法的一种高级实现，它采用更精确的近似算法来寻找最佳的树模型。与传统的梯度增强方法相比，XGBoost有3种主要的不同之处。首先，XGBoost查看叶节点中所有数据点的特征分布，并根据前面的信息减少可能的特征分割的搜索空间。传统的梯度增强树模型通常考虑所有可能的分割组合。相比，XGboost的特征分割方法更有效。其次，XGBoost包含了用于高级正则化的各种超参数，因此它可以更好地阻止模型过度拟合，而我们之前的梯度增强只能通过控制学习率

来解决这个问题。最后，由于XGBoost的核心算法是可并行的，因此计算成本较低，可以应用于较大的数据集。尽管这些优点有时有助于XGBoost胜过许多其他算法，但它们并不能保证在每种情况下都有更好的性能，因为每种模型的错误率严重依赖于特定的数据集。为了保持我们的模型和度量标准的一致性，我们为XGBoost应用相同的调优参数，并在相同的学习速率范围(0.01到0.1)检查它的性能。因此，最优学习率也为0.07，最佳RMSE为0.41529。

2.4模型选择

从下表中，梯度增强回归器给出了最佳的测试RMSE。如果找不到额外的特征信息，我们应该使用梯度增强回归器来做定价预测模型。

Cross-validation RMSEs for different regressors	
Regressors	RMSEs
Lasso	0.44915
Ridge	0.44911
Random Forest	0.48710
Gradient Boosting	0.40907
XG Boosting	0.41529

表一、模型选择表

3结论

像Airbnb这样的在线租赁市场正在使用智能定价模型来提供参考价格。而我们正试图基于Airbnb纽约公开数据建立一个智能定价模型。在我们的建模过程中，我们注意到线性模型往往不适合我们的数据集。为了解决欠拟合问题，我们使用了几个更灵活的模型:随机森林、梯度增强和XGBoost。结果表明，梯度增强回归器产生的测试误差最小。因此，对于当前的数据集，应该使用梯度增强算法来更好地参考市场价格。

4潜在的改进

4.1添加特征

在数据集上使用了更灵活的模型之后，我们在回归误差上得到了一些改进。然而，我们使用的模型中最灵活的模型，即随机森林回归模型，在我们的数据集上表现不佳。如前所述，性能低下的一个主要原因是特性数量不足。这也是这个项目最重要的关注点。因此，我们未来要改进的首要工作是找到更多与地理相关的数据，例如犯罪数据、生活水平数据和交通数据，这样就可以获得更多关于房源的完整信息。

4.2字嵌入

另一个潜在的改进是应用字嵌入来评估每个关键字与房源租赁价格的相关性。引入这些额外的特性也是我们可以对当前数据集进行的最后一步探索。一般来说，如果我们想要更好的预测结果，我们需要更多的与租赁价格相关的特征信息的附加数据。