



项目

Predicting Boston Housing Prices

此部分属于 Machine Learning Engineer Nanodegree Program

项目审阅

代码审阅

注释

与大家分享你取得的成绩！

Meets Specifications



恭喜你通过了这个项目！

接下来的内容也非常有趣，加油！

分析数据

请求的所有 Boston Housing 数据集统计数据均已得到精确计算。学生可恰当利用 NumPy 功能获得这些结果。

非常棒，你用 NumPy 计算出了正确的值！

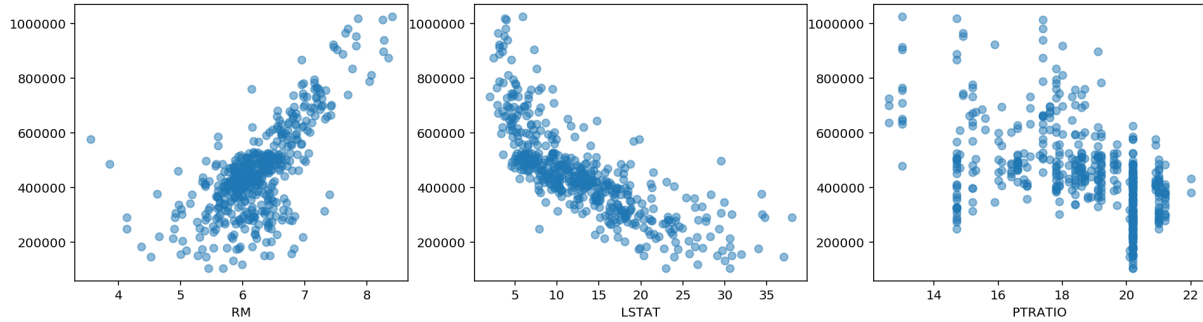
学生正确解释各项属性与目标变量增加或减少之间的关联。

有一种非常好的思路是通过绘制散点图可视化数据来验证自己的猜想，可以参考下面的代码：

```
# 载入画图所需要的库 matplotlib
import matplotlib.pyplot as plt

# 使输出的图像以更高清的方式显示
%config InlineBackend.figure_format = 'retina'
```

```
# 调整图像的宽高
plt.figure(figsize=(16, 4))
for i, key in enumerate(['RM', 'LSTAT', 'PTRATIO']):
    plt.subplot(1, 3, i+1)
    plt.xlabel(key)
    plt.scatter(data[key], data['MEDV'], alpha=0.5)
```



从上图中我们可以看出，虽然‘PTRATIO’与‘MEDV’没有像‘RM’、‘LSTAT’那么强的相关性，但是更倾向于‘PTRATIO’增大‘MEDV’减小，可能的原因是在美国较好的学校会有低的学生老师比，这也是教育资源和教学质量的保证。

模型衡量标准

学生正确判断假设模型是否能根据其 R^2 分数成功捕捉目标变量的方差。

学生合理解释为何要为某个模型将数据集分解为训练子集和测试子集。训练和测试分解会在代码中正确实现。

正确实现了代码。通过比较模型在测试集和训练集的表现，我们能够知道模型是否出现了过拟合或者欠拟合，测试集评估模型在未知数据的泛化能力。

当模型出现过拟合时，模型可能在训练集上表现很好，但却不能泛化到新的数据。

分析模型的表现

随着训练点的不断增加，学生正确判断图表中训练集和验证集曲线的走向并讨论该模型是否会得益于更多的训练点。

很好，你对曲线走势的观察是正确的。

在传统机器学习方法中，当模型学习到一定程度以后，增大训练数据是很难有提升的，但是在深度学习中往往是一种有效的方法。

学生提供最大深度为 1 和 10 的分析。如果模型偏差或方差较高，请针对每个图形给出合理的理由。

很好的判断，高偏差通常是由于模型太简单（即模型欠拟合），不能很好的拟合测试集，训练分数、验证分数、测试分数通常都比较低；高方差通常是由于模型过于复杂（即模型过拟合），模型在训练集上表现得很好，在验证集和测试集上得分确比较低，泛化能力差。

学生根据合理的理由使用模型复杂度图形猜测最优模型的参数。

评估模型性能

学生准确说明网格搜索算法，并简要探讨该算法的用途。

你还可以注意到这一点：网格搜索也不一定总是选择得分最高的模型，如果评分标准是loss，会返回评分最低。GridSearchCV的[文档](#)中也给出了详细说明；Estimator that was chosen by the search, i.e. estimator which gave highest score (or smallest loss if specified) on the left out data.

学生准确说明如何对模型进行交叉验证，以及它对网格搜索的作用。

K折交叉验证并不一定会将数据随机划分，默认是按顺序分，你可以通过查看[源码](#)来确定这一点。

学生在代码中正确实现 `fit_model` 函数。

正确实现了 `fit_model` 函数。

学生根据参数调整确定最佳模型，并将此模型的参数与他们猜测的最佳参数进行对比。

进行预测

学生报告表格所列三位客户的预测出售价格，根据已知数据和先前计算出的描述性统计，讨论这些价格是否合理。

学生计算了最优模型在测试集上的决定系数，并给出了合理的分析。

学生可以合理分析最优模型是否具有健壮性。

在这里10次预测是对同一间房屋的，预测值随训练数据的变化而变化。当Range in prices较小或者在能接受的范围
内就可以认为模型是足够健壮来保证预测的一致性。

学生深入讨论支持或反对使用他们的模型预测房屋售价的理由。

 [下载项目](#)

[返回 PATH](#)

[学员 FAQ](#)