问题构建（Framing）：

什么是（监督式）机器学习：

- 通过学习*如何组合输入信息*来对从未见过的数据做出有用的**预测**。
  - Labels（标签）：指我们要预测的真实事物：**y**
    - 基本线性回归中的y变量。
  - Features（特征）：表示数据的方式，指用于描述数据的输入变量：x_i。
    - 基本线性回归中的$x_1, x_2, x_3, \ldots, x_n$。
  - Sample(样本)：指数据的特定实例（为一个矢量）：**x**.
    - 有标签：具有{特征，标签}：（**x**，**y**）：用于训练模型。
    - 无标签：具有{特征，？}：（**x**，？）：用于对新数据做出预测。
  - Model（模型）：将样本映射到预测标签：**y'**
    - **执行预测的工具**。通过从数据债学习规律这一过程来尝试创建模型。
    - 由模型的内部参数定义，这些内部参数值是通过学习得到的。

深入了解机器学习（Descending into ML）：

线性回归：是一种找到最合适一组点的直线或超平面的方法。

$y = wx + b$

w为权重矢量（直线斜率）。b为偏差。

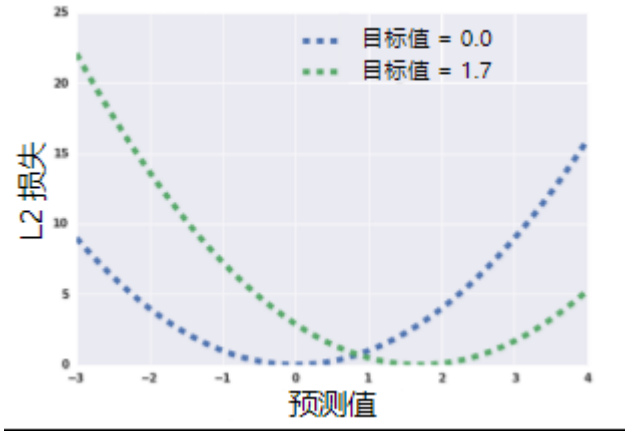
误差（loss ）（针对单个样本，采用模型的预测结果与真实值之间的方差）：

给定样本的 **L2 损失**也称为平方误差

= 预测值和标签值之差的平方

= (观察值 - 预测值)2

= (y - y')2



定义数据集上的 L2 损失(方差)

$$L_2 Loss = \sum_{(x,y)\in D}(y - prediction(x))^2$$

不是专注于减少某一个样本，而是着眼于最大限度地减少整个数据集的误差。

训练与损失

训练模型表示通过有标签的样本来学习（确定）所有权重和偏差的理想值。在监督学习中，ML通过以下方式构建模型： *检查多个样本并尝试找出最大限度地减少损失的模型----**经验风险最小化。***

损失是对错误预测的惩罚。他是一个*数值*，表示对于单个样本而言模型预测的准确程度。训练模型的目标是从所有样本中找到一组平均损失“较小”的权重和偏差。

常见损失（平方损失）：

均方误差 (**MSE**) 指的是每个样本的*平均平方损失*。要计算 MSE，需要求出各个样本的所有平方损失之和，然后除以样本数量：

$$MSE = \frac{1}{N}\sum_{(x,y)\in D}(y - prediction(x))^2$$

其中，

- (x, y)指的是样本，其中
  - x指的是模型进行预测时使用的特征集（例如，温度、年龄和交配成功率）。
  - y指的是样本的标签（例如，每分钟的鸣叫次数）（也就是真实值）。
- $prediction(x)$指的是权重和偏差与特征集 x结合的函数（使用模型预测得到的值）。
- D指的是包含多个有标签样本（即 (x, y)）的数据集。

- N指的是 D 中的样本数量。

  虽然 MSE 常用于机器学习，但它既不是唯一实用的损失函数，也不是适用于所有情形的最佳损失函数。

降低损失（Reducing Loss）：

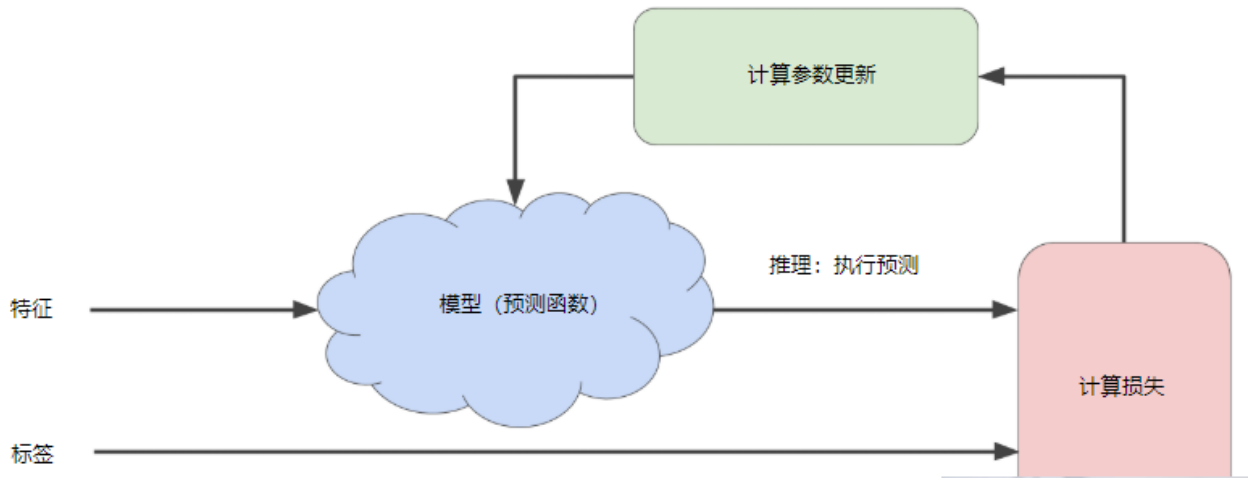
迭代方法是一种广泛用于降低损失的方法，使用起来也简单有效。

如何降低损失?

梯度：与模型参数相关的误差函数的导数。

- $(y-y')2$ 相对于权重和偏差的导数可让我们了解指定样本的损失变化情况
  - 易于计算且为凸形
- 因此，我们在能够尽可能降低损失的方向上反复采取小步
  - 我们将这些小步称为**梯度步长**（但它们实际上是负梯度步长）
  - 这种优化策略称为**梯度下降法**

梯度下降法示意图：



权重初始化:

- 对于凸形问题，权重可以从任何位置开始（比如，所有值为 0 的位置）
  - 凸形：想象一个碗的形状

    - 只有一个最低点
- 预示：不适用于神经网络
    - 非凸形：想象一个蛋托的形状
    - 有多个最低点
    - 很大程度上取决于初始值

SGD 和小批量梯度下降法

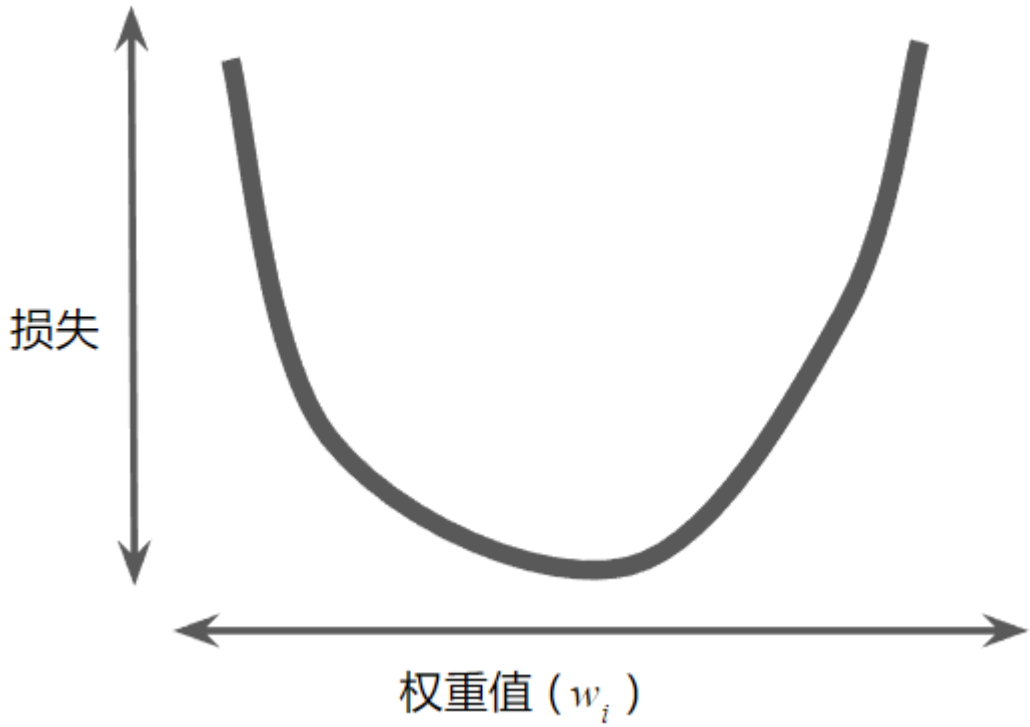
- 可以在每步上计算整个数据集的梯度，但事实证明没有必要这样做
- 计算小型数据样本的梯度效果很好
    - 每一步抽取一个新的随机样本
- **随机梯度下降法**：一次抽取一个样本
- **小批量梯度下降法**：每批包含 10-1000 个样本
    - 损失和梯度在整批范围内达到平衡

*通常，可以不断迭代，直到总体损失不再变化或至少变化极其缓慢为止。这时候，我们可以说该模型已**收敛**。*

要点：

在训练机器学习模型时，首先对权重和偏差进行初始猜测，然后反复调整这些猜测，直到获得损失可能最低的权重和偏差为止。

降低损失 (Reducing Loss)：梯度下降法:



凸形问题只有一个最低点；即只存在一个斜率正好为 0 的位置。这个最小值就是损失函数收敛之处，也可以称这个最低点为**全局最小值**。

但是，通过计算整个数据集中 w1 每个可能值的损失函数来找到收敛点这种方法效率太低。

因此可以研究一种更好的机制，这种机制在机器学习领域非常热门，称为**梯度下降法**。

梯度是偏导数的矢量；它可以让您了解哪个方向距离目标“更近”或“更远”。

梯度是一个矢量，因此具有以下两个特征：

- 方向
- 大小

梯度始终指向损失函数中增长最为迅猛的方向。梯度下降法算法会沿着**负梯度**的方向走一步，以便尽快降低损失。

降低损失 (Reducing Loss)：学习速率:

梯度矢量具有方向和大小。梯度下降法算法用梯度乘以一个称为**学习速率**（有时也称为**步长**）的标量，以确定下一个点的位置。

每个回归问题都存在一个金发姑娘学习速率。“金发姑娘”值与损失函数的平坦程度相关。如果您知道损失函数的梯度较小，则可以放心地试着采用更大的学习速率，以补偿较小的梯度并获得更大的步长 。

一维空间中的理想学习速率是 $\frac{1}{f(x)''}$（f(x) 对 x 的二阶导数的倒数）。

二维或多维空间中的理想学习速率是海森矩阵（由二阶偏导数组成的矩阵）的倒数。

广义凸函数的情况则更为复杂。

降低损失 (Reducing Loss)：随机梯度下降法:

- **批量（batch）**指的是用于在单次迭代中计算梯度的样本总数。
- 包含随机抽样样本的大型数据集可能包含冗余数据。实际上，批量大小越大，出现冗余的可能性就越高。一些冗余可能有助于消除杂乱的梯度，但超大批量所具备的预测价值往往并不比大型批量高。
- 通过从我们的数据集中随机选择样本，我们可以通过小得多的数据集估算（尽管过程非常杂乱）出较大的平均值。
- **随机梯度下降法 (SGD)** 将这种想法运用到极致，它每次迭代只使用一个样本（批量大小为 1）。如果进行足够的迭代，SGD 也可以发挥作用，但过程会非常杂乱。“随机”这一术语表示构成各个批量的一个样本都是随机选择的。
- **小批量随机梯度下降法（小批量 SGD）**是介于全批量迭代与 SGD 之间的折衷方案。小批量通常包含**10-1000**个随机选择的样本。小批量 SGD 可以减少 SGD 中的杂乱样本数量，但仍然比全批量更高效。

有适用于模型调整的标准启发法吗？ 这是一个常见的问题。简短的答案是，**不同超参数的效果取决于数据。**因此，不存在必须遵循的规则，您需要对自己的数据进行测试。

即便如此，我们仍在下面列出了几条可为您提供指导的经验法则：

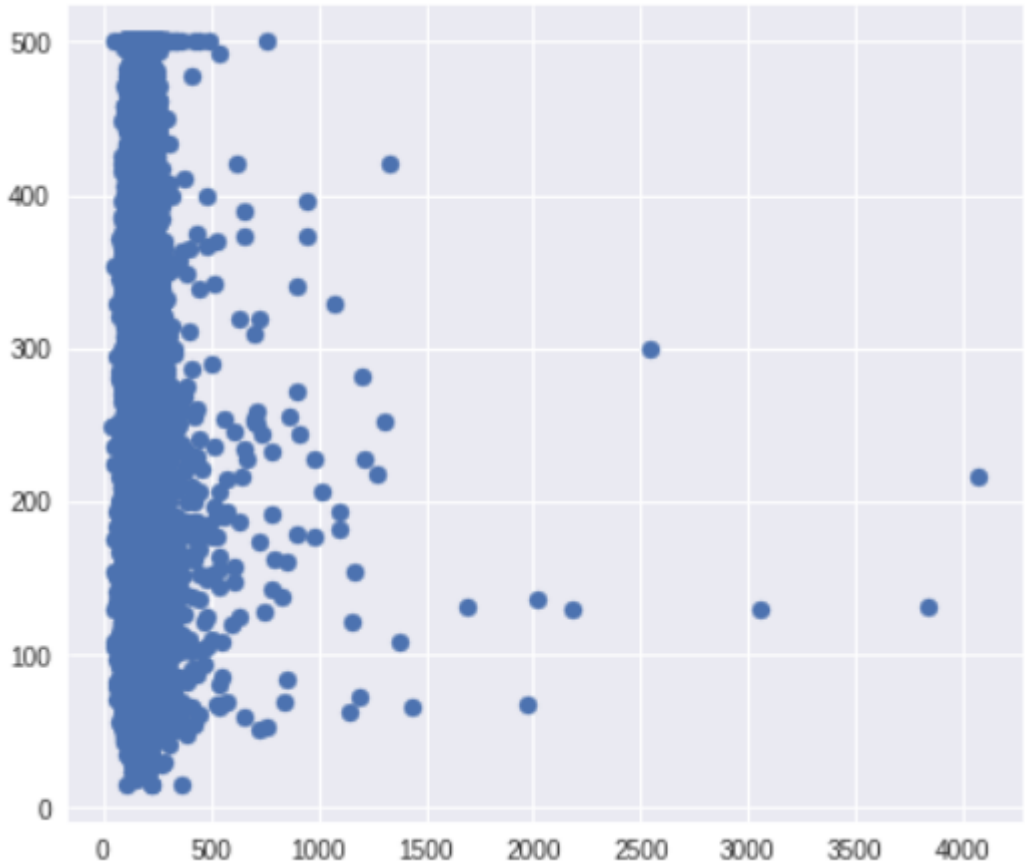
- 训练误差应该稳步减小，刚开始是急剧减小，最终应随着训练收敛达到平稳状态。
- 如果训练尚未收敛，尝试运行更长的时间。
- 如果训练误差减小但速度过慢，则提高学习速率也许有助于加快其减小速度。

    - 但有时如果学习速率过高，训练误差的减小速度反而会变慢。
- 如果训练误差变化很大，尝试降低学习速率。
    - 较低的学习速率和较大的步数/较大的批量大小通常是不错的组合。*
- 批量大小过小也会导致不稳定情况。不妨先尝试 100 或 1000 等较大的值，然后逐渐减小值的大小，直到出现性能降低的情况。

重申一下，切勿严格遵循这些经验法则，因为效果取决于数据。请始终进行试验和验证。

识别离群值：

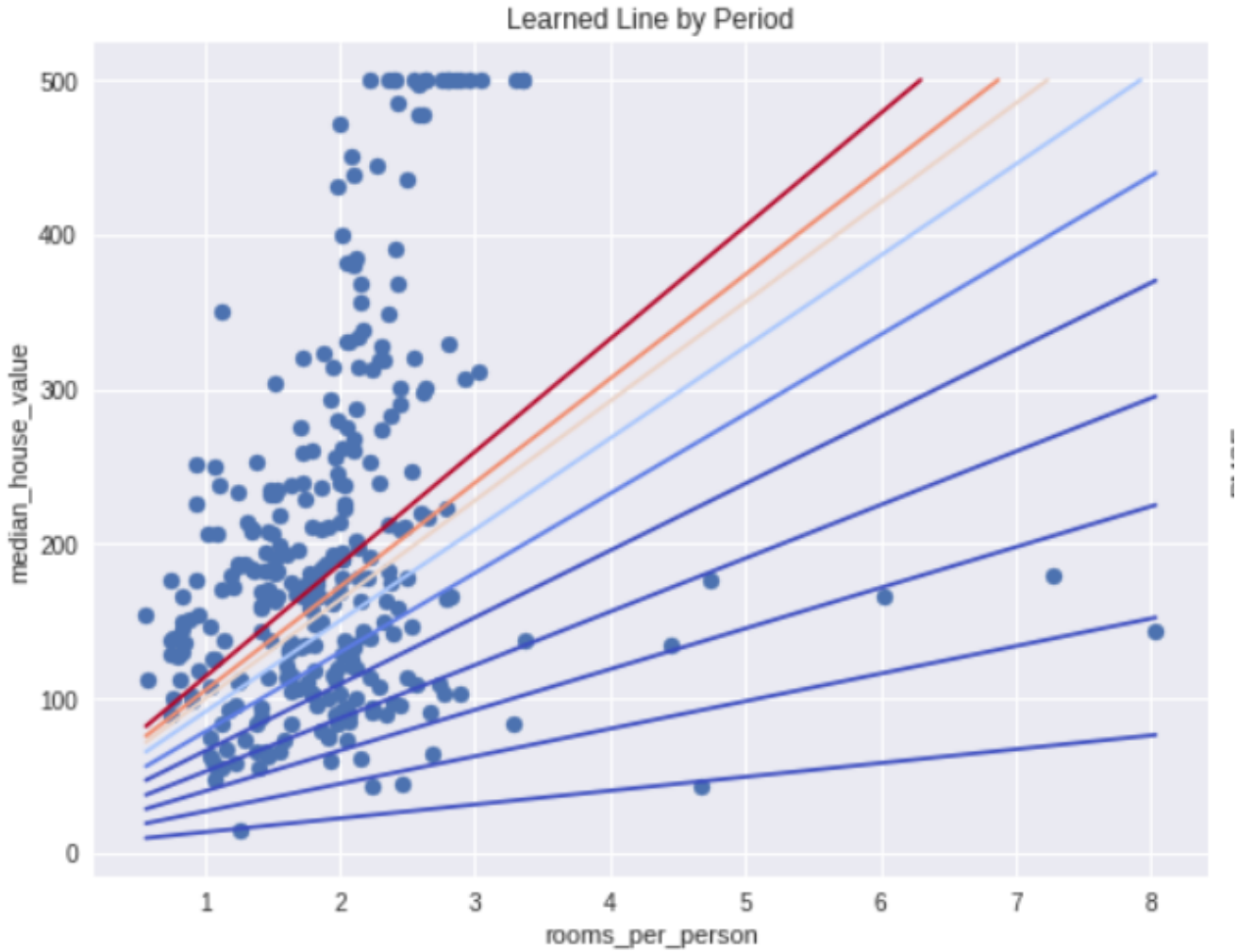
在源数据中可能会存在一些异常数据影响模型的预测结果，可以称这些数据为**离群值**，也可说是偏离了源数据分布的一些值。



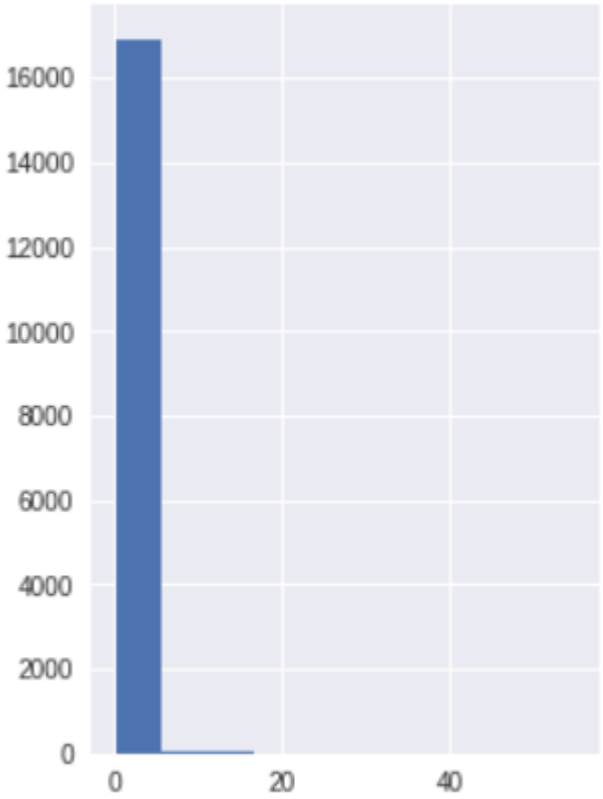
如上图右侧的个别点。由于点相对数量较小，故判断他们为离群值。

可以通过将这些离群值设置为相对合理的最小值或最大值来改进模型拟合情况。

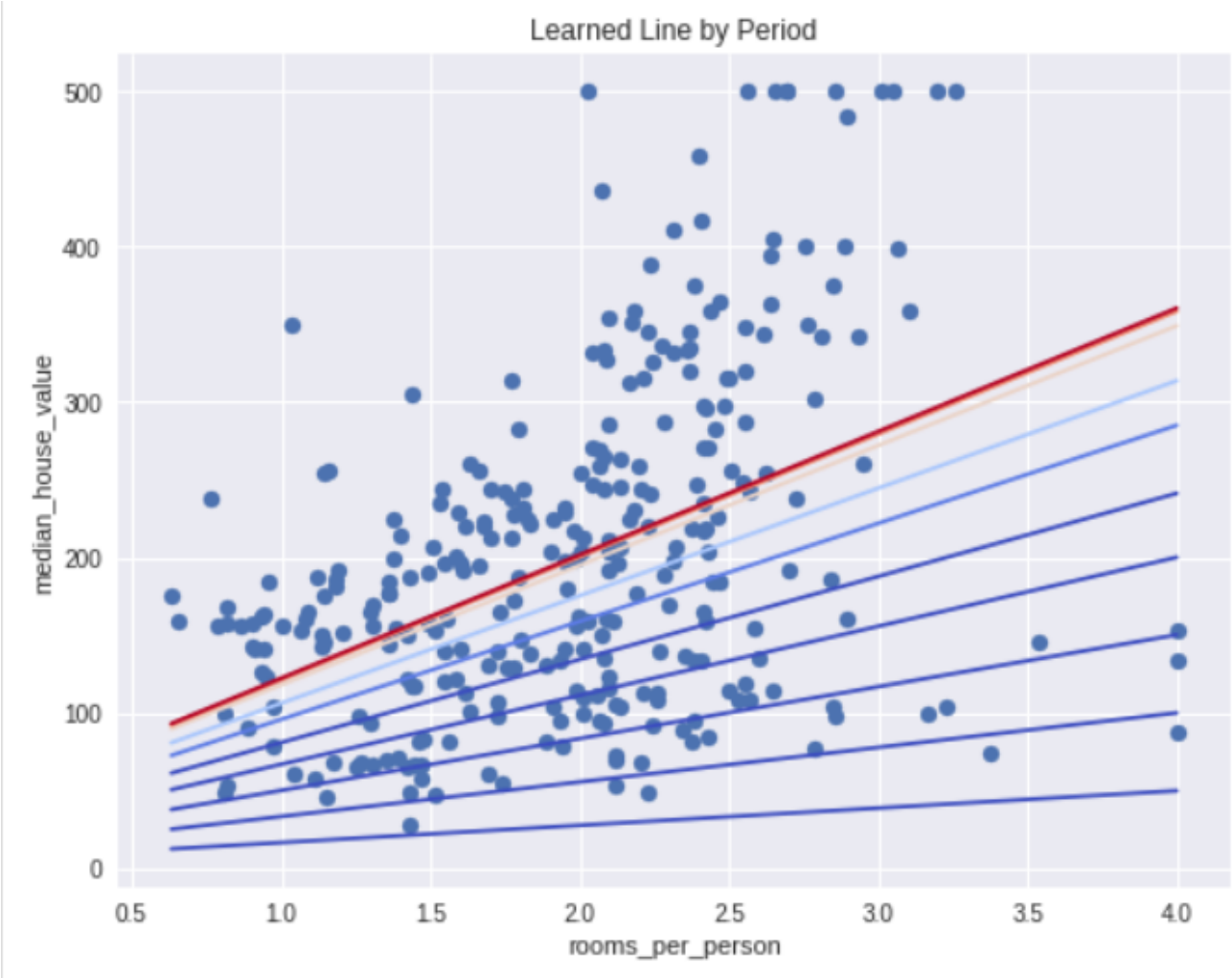
例如初始情况，rooms_per_person的范围为0-8.右侧有几个离群值点：



通过下述rooms_per_person的直方图所示，发现大多数值都小于5，故将值截取为5.



在截取为5后的结果：



会发现少了那些离群值的影响，最后的数据分布看起来更加容易通过我们的模型来进行拟合。