树回归

1. 一些我对树回归的理解

类似排列组合中的插值法，树回归的任务完成后，就相当于是一个数轴被划成了好几部分，每一部分取均值来作为刻度。

我对回归的理解是，一些数据回归到一个函数上去，树回归不像线性回归，树回归最终处理出来的结果是分段的，线性回归最终处理出来的结果是连续的。

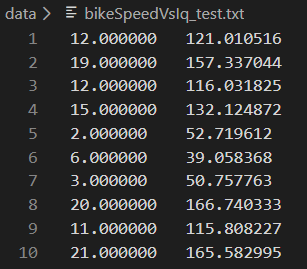
1. 代码处理

划分函数为：

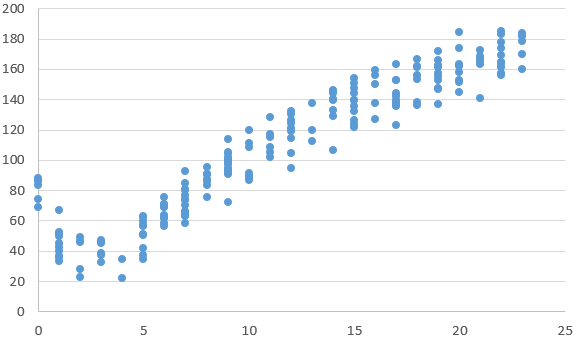
|  |  |
| --- | --- |
| 函数名 | 功能 |
| binSplitDataSet | 根据某列中的某值切分数据集 |
| regLeaf | 求均值 |
| regErr | 求总方差，这里为方差乘上数量 |
| chooseBestSplit | 选择特征索引和最佳值 |
| createTree | 递归构建树，返回存储树的字典 |

1. 实验结果与结论

数据集用的是一个自行车速的数据集，

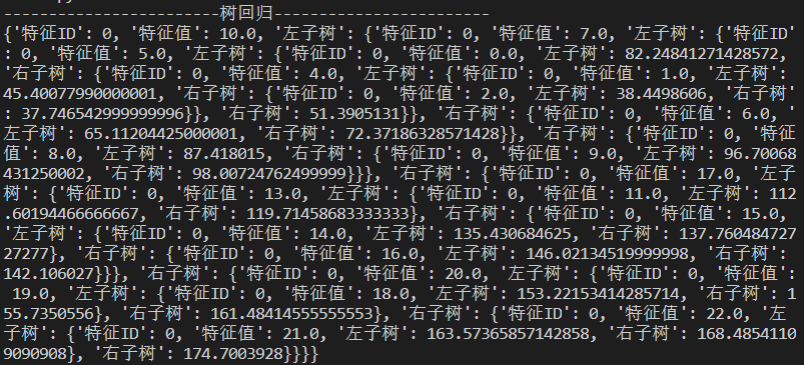


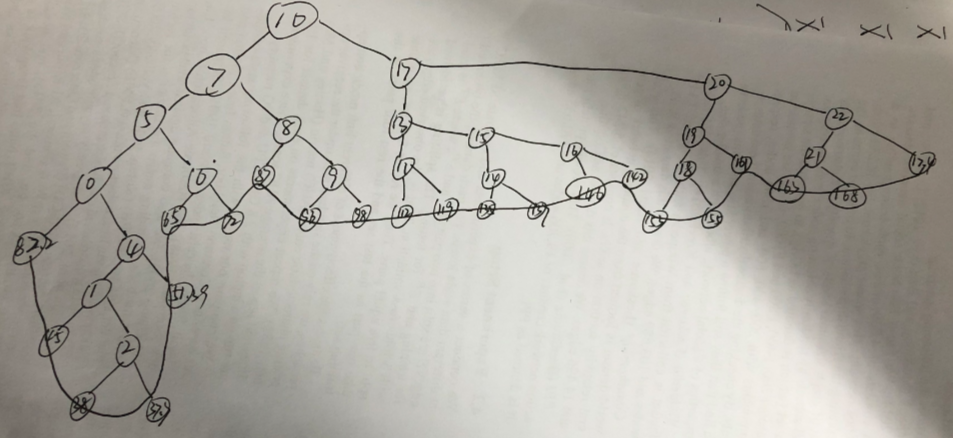
在运行程序之前现在Excel中画一下图看看数据是什么样的。



根据图像，不出意外的话最终会有24个叶子节点。

运行程序：



数了一下个数发现真的就是24个叶子节点。并且把树画出来，基本拟合原图的单调性。

不知道是不是树回归就真的有这样的一种属性，就是把数据模拟成原函数的图像，只不过是分段的、阶梯的。我在想加入数据足够松散，是不是就能完全拟合（过拟合）。当然这种情况应该不太现实，毕竟下一小节就是树的剪枝了，树的剪枝应该就是防止过拟合的。

至于运行出来的结果中特征ID老是0的现象，这个是我认为与ID3和C4.5不同的地方，ID3和C4.5的算法是每次切分就少一个特征，而树回归可以一直在一个特征上切，只要没达到阈值条件就能一直切下去。

1. 总结
2. 树回归中选择最佳特征和切分点的流程：

* 循环遍历每一个特征的特征值，按值切分为两个子集，根据Y找最小的误差；
* 标记出特征及其对应的值然后切分
* 重复此过程直到满足设定的条件

1. 建树仍然是用递归
2. ID3特征选取准则是信息增益，C4.5特征选取准则是信息增益比，CART做回归的特征选取准则是总方差。
3. 关于剪枝：预剪枝：在chooseBestSplit函数中的几个提前终止条件(切分样本小于阈值、混乱度减弱小于阈值)，都是预剪枝（参数敏感）。