开课吧-小钟-20191020

笔记本:	开课吧-小钟讲课		
创建时间:	2019/10/11 星期五 13:43	**更新时间:**	2019/10/19 星期六 21:47
作者:	你看起来好像很好吃n_n		
URL:	https://baike.baidu.com/item/Python/407313?fr=aladdin		

开课吧-数据竞赛及相关问题从小工到专家

1.1python 介绍-磨刀不误砍柴工

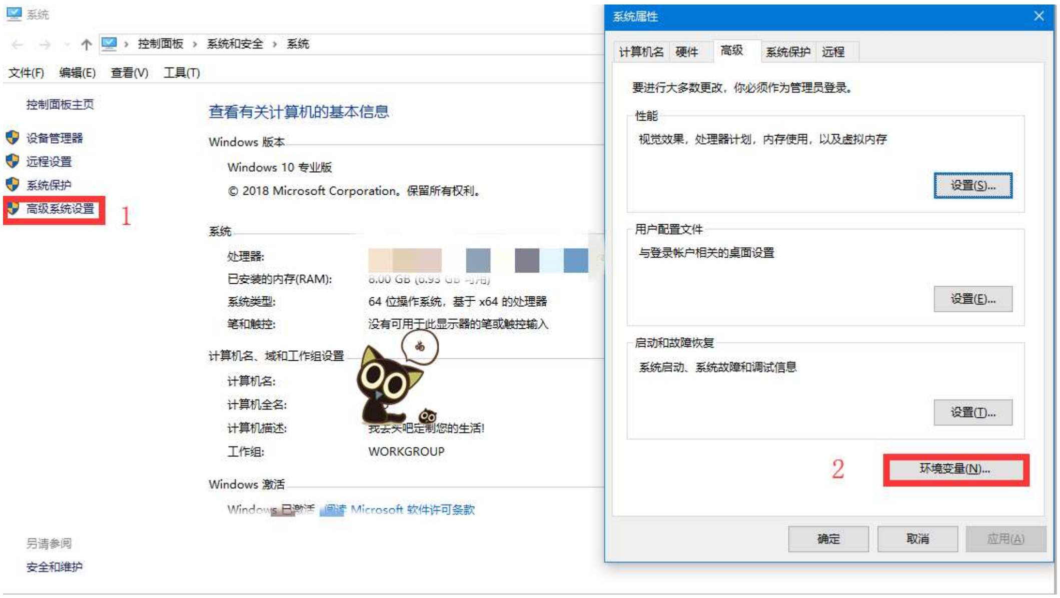
大体框架

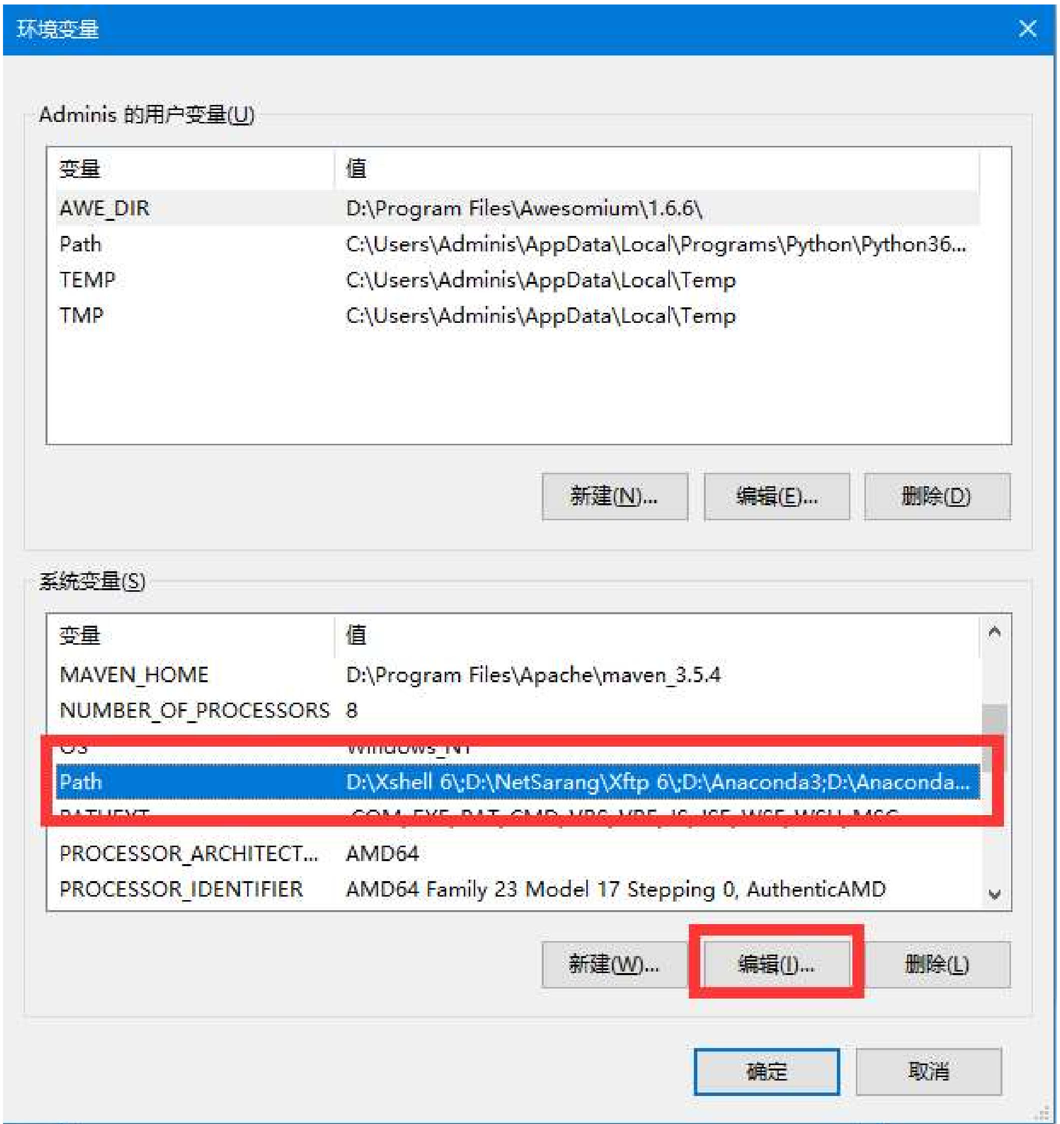
- 语言：python
- 比赛：国内竞赛平台，kaggle
- 类别包括：二分类，多分类，回归，时序
- 项目业务类型：反欺诈，信用评估，工业项目，金融风控，

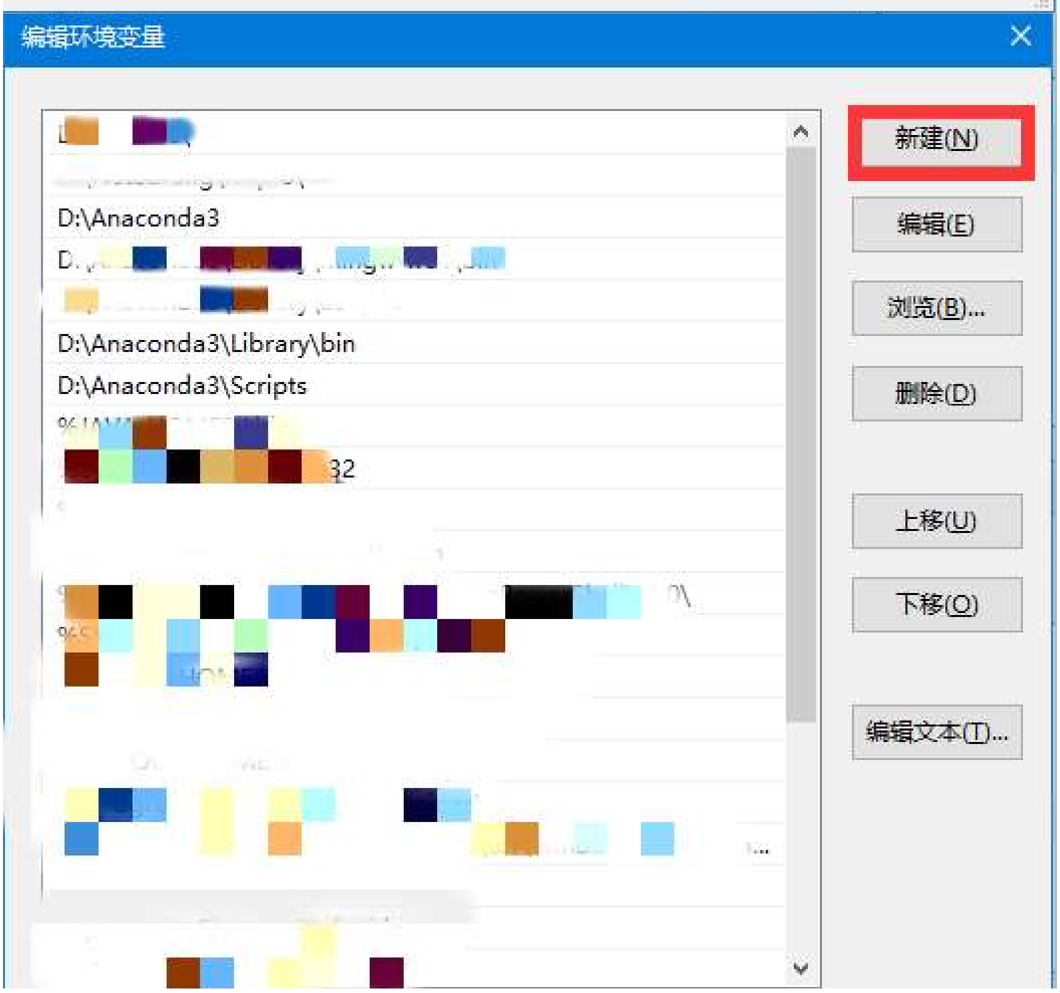
1. Python （计算机程序设计语言）

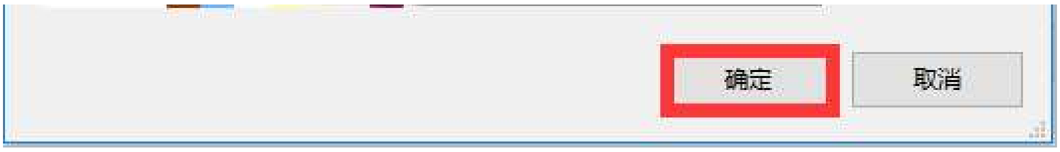
Python是一种跨平台的计算机程序设计语言。是一种面向对象的动态类型语言，最初被设计用于编写自动化脚本(shell)，随着版本的不断更新和语言新功能的添加，越来越多被用于独立的、大型项目的开发。

2. python 官网
3. Aanaconda
4. Anaconda清华镜像
5. Anaconda安装及其环境变量配置：这里是windows10
5.1 D:\Anaconda3
5.2 D:\Anaconda3\Scripts
5.3 D:\Anaconda3\Library\bin


系统
控制面板 › 系统和安全 › 系统
文件(F) 编辑(E) 查看(V) 工具(T)
控制面板主页
设备管理器
远程设置
系统保护
高级系统设置
1
查看有关计算机的基本信息
Windows 版本
Windows 10 专业版
© 2018 Microsoft Corporation。保留所有权利。
系统
处理器:
已安装的内存(RAM):
系统类型:
64 位操作系统，基于 x64 的处理器
笔和触控:
没有可用于此显示器的笔或触控输入
计算机名、域和工作组设置
计算机名:
计算机全名:
计算机描述:
工作组:
WORKGROUP
Windows 激活
Microsoft 软件许可条款
另请参阅
安全和维护
系统属性
计算机名 硬件 高级 系统保护 远程
要进行大多数更改，你必须作为管理员登录。
性能
视觉效果，处理器计划，内存使用，以及虚拟内存
设置(S)...
用户配置文件
与登录帐户相关的桌面设置
设置(E)...
启动和故障恢复
系统启动、系统故障和调试信息
设置(T)...
2
环境变量(N)...
确定
取消
应用(A)

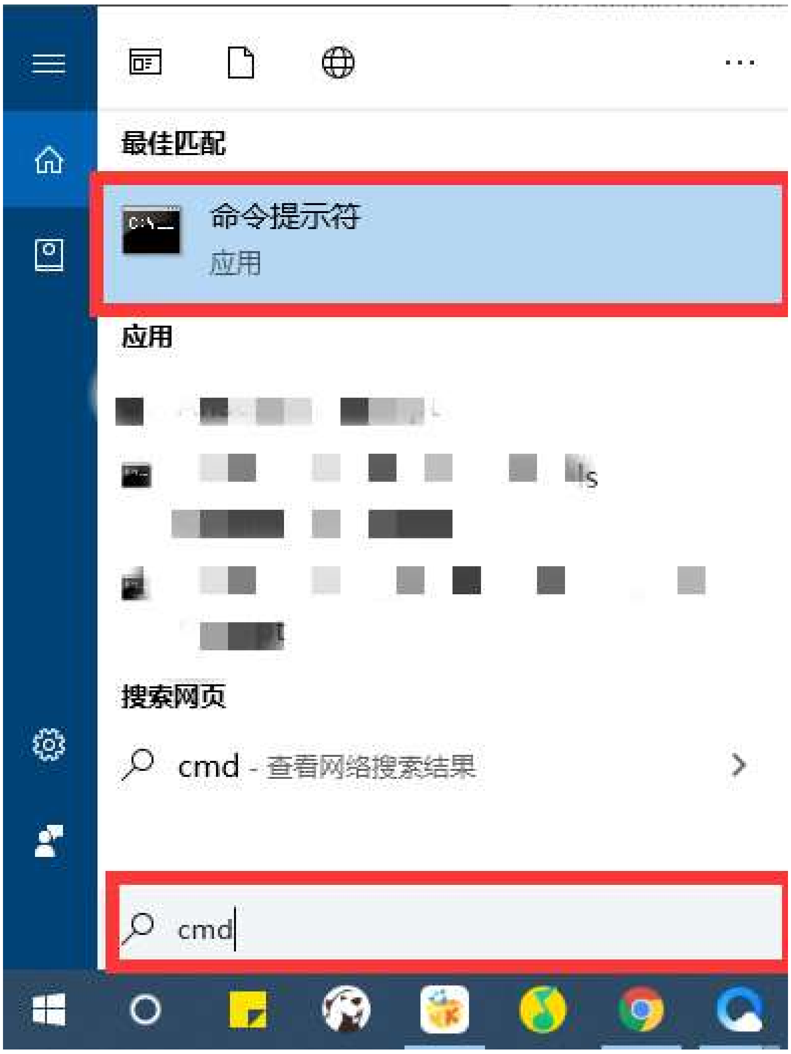

环境变量
Adminis 的用户变量(U)
变量
值
AWE_DIR
D:\Program Files\Awesomium\1.6.6\
Path
C:\Users\Adminis\AppData\Local\Programs\Python\Python36...
TEMP
C:\Users\Adminis\AppData\Local\Temp
TMP
C:\Users\Adminis\AppData\Local\Temp
新建(N)...
编辑(E)...
删除(D)
系统变量(S)
变量
值
MAVEN_HOME
D:\Program Files\Apache\maven_3.5.4
NUMBER_OF_PROCESSORS
8
Path
D:\Xshell 6\;D:\NetSarang\Xftp 6\;D:\Anaconda3;D:\Anaconda...
PROCESSOR_ARCHITECT...
AMD64
PROCESSOR_IDENTIFIER
AMD64 Family 23 Model 17 Stepping 0, AuthenticAMD
新建(W)...
编辑(I)...
删除(L)
确定
取消


编辑环境变量
D:\Anaconda3
D:\Anaconda3\Library\bin
D:\Anaconda3\Scripts
新建(N)
编辑(E)
浏览(B)...
删除(D)
上移(U)
下移(O)
编辑文本(T)...



6.安装所需要的包

在windows当中主要通过pip install 来下载我们所需要的包



7.初次使用Jupyter-Notebook

8.介绍一些常用的包

```
import os
import json
import gc

from numba import jit
#tqdm
# os.system('pip install tqdm')
```

```
from tqdm import tqdm_notebook
from tqdm import tqdm

#Integrated model
# os.system('pip install lightgbm')
import lightgbm as lgb
# os.system('pip install catboost==0.15.2')
import catboost as cbt
# os.system('pip install xgboost')
# import xgboost as xgb

#base import
import numpy as np
import pandas as pd

# about sklearn
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import StratifiedKFold, KFold,
RepeatedKFold
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler as std
from sklearn.kernel_ridge import KernelRidge
from sklearn.metrics import f1_score
#about time
import time
import datetime
from datetime import datetime, timedelta

#Garbage collection
import gc
#other
from collections import Counter
from statistics import mode
   #warning
import warnings
warnings.filterwarnings("ignore")
import json
import math
tqdm.pandas()
os.system('pip install re')
import re
```

1.2国内外常用竞赛网站介绍

1. 天池
   1.1天池竞赛
   1.2天池AI学习
   1.3天池一些规则
2. DataFountain.
   2.1DataFountain一些组队规则，相关经验介绍
3. 科赛网
4. DC大赛
5. kaggle

1.3两个常用集成决策树模型原理介绍

1. xgboost 原理
1.1xgboost原始论文地址
1.2xgboost 原始ppt介绍
1.3理解xgboost 所需基础（xgboost有很多cart）
CART(clasification and regression tree):在给定输入随机变量X条件下输出随机变量Y的条件概率分布的学习方法：
stpe1:决策树生成：基于训练数据生成决策树，生成的决策树要尽量大；
step2:用验证数据集对已经生成的树进行剪枝并选择最优子树，这时用损失函数最小作为剪枝的标准。
分类树和回归树的区别
1.3.1分类树使用信息增益或增益比率来划分节点；每个节点样本的类别情况投票决定测试样本的类别。
1.3.2回归树使用最大均方差划分节点；每个节点样本的均值作为测试样本的回归预测值。
1.3.3基尼系数，又叫基尼不纯度，表示样本集合中被随机选中的一个样本被错误分类的概率，值越小表示被分错的概率越小，**基尼指数**=被选中的概率*被分错的概率，如下公式中， pk表示选中的样本属于k类别的概率，则这个样本被分错的概率是(1-pk)（李航《统计学习方法-第一版》69）

定义 5.4（基尼指数） 分类问题中，假设有 K 个类，样本点属于第 k 类的概率为 p_k，则概率分布的基尼指数定义为

$$\mathrm{Gini}(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2 \tag{5.22}$$

如果样本集合 D 根据特征 A 是否取某一可能值 a 被分割成 D_1 和 D_2 两部分，即

$$D_1=\{(x,y)\in D \mid A(x)=a\}, \quad D_2=D-D_1$$

则在特征 A 的条件下，集合 D 的基尼指数定义为

$$\mathrm{Gini}(D,A)=\frac{|D_1|}{|D|}\mathrm{Gini}(D_1)+\frac{|D_2|}{|D|}\mathrm{Gini}(D_2) \tag{5.25}$$

基尼指数 $\mathrm{Gini}(D)$ 表示集合 D 的不确定性，基尼指数 $\mathrm{Gini}(D,A)$ 表示经 $A=a$ 分割后集合 D 的不确定性. 基尼指数值越大，样本集合的不确定性也就越大，这一点与熵相似.

表 5.1 贷款申请样本数据表

ID	年龄	有工作	有自己的房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	否	是	非常好	是
10	中年	否	是	非常好	是
11	老年	否	是	非常好	是
12	老年	否	是	好	是
13	老年	是	否	好	是
14	老年	是	否	非常好	是
15	老年	否	否	一般	否

例 5.4 根据表 5.1 所给训练数据集，应用 CART 算法生成决策树.

解 首先计算各特征的基尼指数，选择最优特征以及其最优切分点．仍采用例 5.2 的记号，分别以 A_1，A_2，A_3，A_4 表示年龄、有工作、有自己的房子和信贷情况 4 个特征，并以 1，2，3 表示年龄的值为青年、中年和老年，以 1，2 表示有工作和有自己的房子的值为是和否，以 1，2，3 表示信贷情况的值为非常好、好和一般.

求特征 A_1 的基尼指数:

$$\mathrm{Gini}(D,A_1=1)=\frac{5}{15}\left(2\times\frac{2}{5}\times\left(1-\frac{2}{5}\right)\right)+\frac{10}{15}\left(2\times\frac{7}{10}\times\left(1-\frac{7}{10}\right)\right)=0.44$$

$$\mathrm{Gini}(D,A_1=2)=0.48$$

$$\mathrm{Gini}(D,A_1=3)=0.44$$

由于 $\mathrm{Gini}(D,A_1=1)$ 和 $\mathrm{Gini}(D,A_1=3)$ 相等，且最小，所以 $A_1=1$ 和 $A_1=3$ 都可以选作 A_1 的最优切分点.

求特征 A_2 和 A_3 的基尼指数：

$$\mathrm{Gini}(D,A_2=1)=0.32$$

$$\mathrm{Gini}(D,A_3=1)=0.27$$

由于 A_2 和 A_3 只有一个切分点，所以它们就是最优切分点.

求特征 A_4 的基尼指数：

$$\mathrm{Gini}(D,A_4=1)=0.36$$

$$\mathrm{Gini}(D,A_4=2)=0.47$$

$$\mathrm{Gini}(D,A_4=3)=0.32$$

$\mathrm{Gini}(D,A_4=3)$ 最小，所以 $A_4=3$ 为 A_4 的最优切分点.

在 A_1，A_2，A_3，A_4 几个特征中，$\mathrm{Gini}(D,A_3=1)=0.27$ 最小，所以选择特征 A_3 为最优特征，$A_3=1$ 为其最优切分点. 于是根结点生成两个子结点，一个是叶结点. 对另一个结点继续使用以上方法在 A_1，A_2，A_4 中选择最优特征及其最优切分点，结果是 $A_2=1$．依此计算得知，所得结点都是叶结点. ■

对于本问题，按照 CART 算法所生成的决策树与按照 ID3 算法所生成的决策树完全一致.

回归树

ex.(最下二乘法回归树生成算法)

输入：训练数据集*D*;

输出：回归树*f*(x).

在训练数据集所在的输入空间中，递归地将每个区域划分为两个子区域并决定每个子区域上的输出值，构建二叉决策树。

（1）选择最优切分变量j与切分点s，求解

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

遍历变量j，对固定的切分变量j扫描切分点s，选择使上式达到最小值的对(j,s)j。

（2）用选定的对(j,s)划分区域并决定响应的输出值：

$$R_1(j,s)=\left\{x \middle| x^{(j)} \le s\right\}, R_2(j,s)=\left\{x \middle| x^{(j)} > s\right\}$$

$$\hat{c}_m=\frac{1}{N_m}\sum_{x_i\in R_m(j,s)} y_i, x\in R_m, m=1,2$$

（3）继续对两个子区域条用步骤（1），（2），直至满足停止条件。

（4）将输入空间划分为M个区域R1,R2,...,RM
,生成决策树：

$$f(x)=\sum_{m=1}^{M}\hat{c}_m I(x\in R_m)$$

** 回归树的例子：**

x	1	2	3	4	5	6	7	8	9	10
y	5.56	5.70	5.91	6.40	6.80	7.05	8.90	8.70	9.00	9.05

当s=1.5时

$$R_1=\{1\}, R_2=\{2,3,4,5,6,7,8,9,10\}$$

$$c_1=5.56, c_2=\frac{1}{9}(5.70+5.91+6.40+6.80+7.05+8.90+8.70+9.00+9.05)=7.50$$

$$m(1.5)=0+15.72=15.72$$

```
(5.56-5.56)**2+ (5.70-7.50)**2+ (5.91-7.50)**2+ (6.40-7.50)**2+ (6.80-
7.50)**2+ (7.05-7.50)**2+ (8.90-7.50)**2+ (8.70-7.50)**2+ (9.00-
7.50)**2+ (9.05-7.50)**2
```

s	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5
c1	5.56	5.63	5.72	5.89	6.07	6.24	6.62	6.88	7.11
c2	7.5	7.73	7.99	8.25	8.54	8.91	8.92	9.03	9.05
m(s)	15.72	12.07	8.36	5.78	3.91	1.93	8.01	11.73	15.74

在表中我们可以发现s=6.5时，c1=6.24,c2=8.91,m(s)最小。因此j=x,s=6.5,回归树f1(x)：

$$f_1(x)=\begin{cases}6.24, x\le 6.5\\ 8.91, x>6.5\end{cases}$$

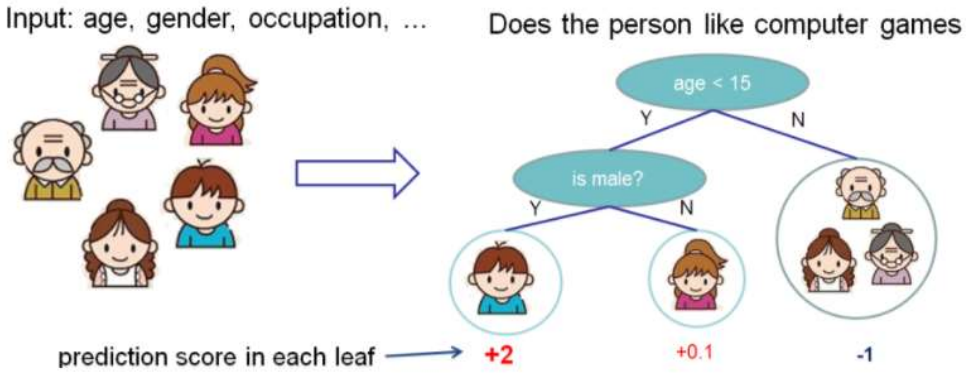
对x≤6.5部分进行划分，回归树f2(x)：

$$f_2(x) = \begin{cases} 5.72, x \leq 3.5 \\ 6.75, 3.5 < x \leq 6.5 \\ 8.91, x > 6.5 \end{cases}$$

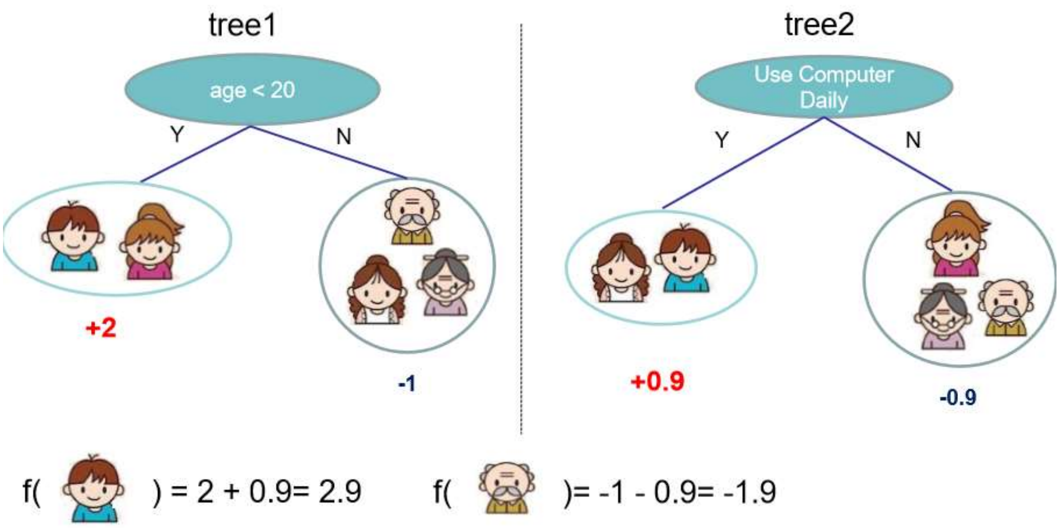
依此类推x>6.5部分，之后不断重复直到满足条件（树的深度、树的叶子个数等都可以作为停止条件）

1.4 xgboost

单颗决策树：



集成思想



逻辑回归和线性回归的表达式：

$$\hat{y}_i = \sum_j w_j x_{ij}$$

逻辑回归需要加上1/(1+exp(-y))

目标函数：

$$Obj(\Theta) = L(\Theta) + \Omega(\Theta)$$

Square loss: $l(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$

Logistic loss: $l(y_i, \hat{y}_i) = y_i \ln(1 + e^{-\hat{y}_i}) + (1 - y_i) \ln(1 + e^{\hat{y}_i})$

一个xgboost ensemble model使用K个累加的函数来预测输出：

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i)$$

表示K个累加函数预测输出

最后最小化正则化目标

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

$$\text{where}\ \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2$$

加法训练过程（从第一步开始优化-一直到最后一步）：

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\dots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

通过上次加法过程，目标函数转变为：

目标函数：$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant$

使用泰勒展开式近似原来的目标函数：

泰勒展开式：

$$f(x + \Delta x) \simeq f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2$$

定义：

$$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}), \quad h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$$

目标函数转化为：

$$Obj^{(t)} \simeq \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant$$

上一步的结果也是常数，对于优化目标函数并无影响

$$\begin{aligned}
Obj^{(t)} &\simeq \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \\
&= \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2 \right] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2 \\
&= \sum_{j=1}^{T} \left[(\sum_{i \in I_j} g_i) w_j + \frac{1}{2} (\sum_{i \in I_j} h_i + \lambda) w_j^2 \right] + \gamma T
\end{aligned}$$

其中I被定义为每个叶子上面样本集合 $I_j = \{i | q(x_i) = j\}$

定义：

$$G_j=\sum_{i\in I_j} g_i \quad H_j=\sum_{i\in I_j} h_i$$

公式进一步为：

$$Obj^{(t)}=\sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j+\frac{1}{2}(\sum_{i\in I_j} h_i+\lambda)w_j^2]+\gamma T$$
$$=\sum_{j=1}^{T}[G_j w_j+\frac{1}{2}(H_j+\lambda)w_j^2]+\gamma T$$

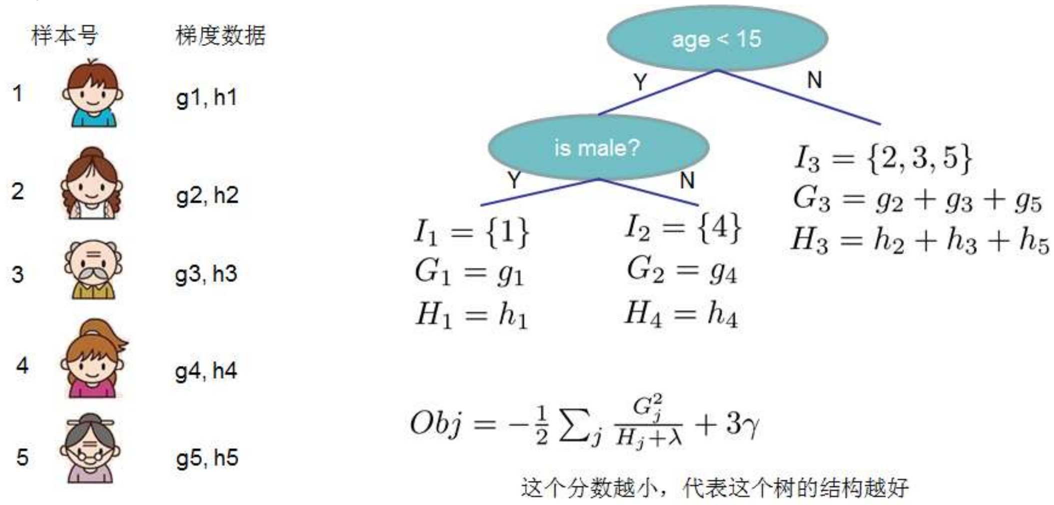
通过对w_j求导等于0：

$$w_j^*=-\frac{G_j}{H_j+\lambda}$$

最后：

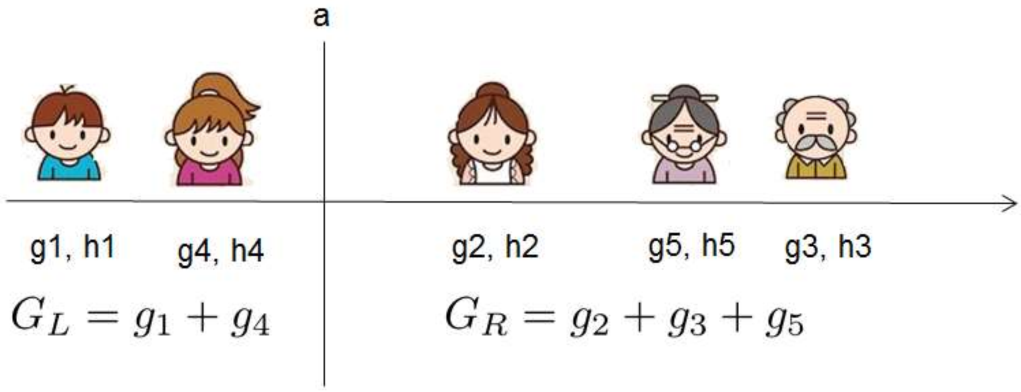
$$Obj=-\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda}+\gamma T$$

Obj示例：



贪心不同树：

$$Gain=\frac{1}{2}[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}]-\gamma$$



2.**lightgbm**

优势

GBDT采用负梯度作为划分的指标（信息增益），XGBoost则利用到二阶导数。GBDT和xgboost 计算信息增益需要扫描所有样本，从而找到最优划分点。在面对大量数据或者特征维度很高时，他们的效率和扩展性很难使人满意。

提高了速度：

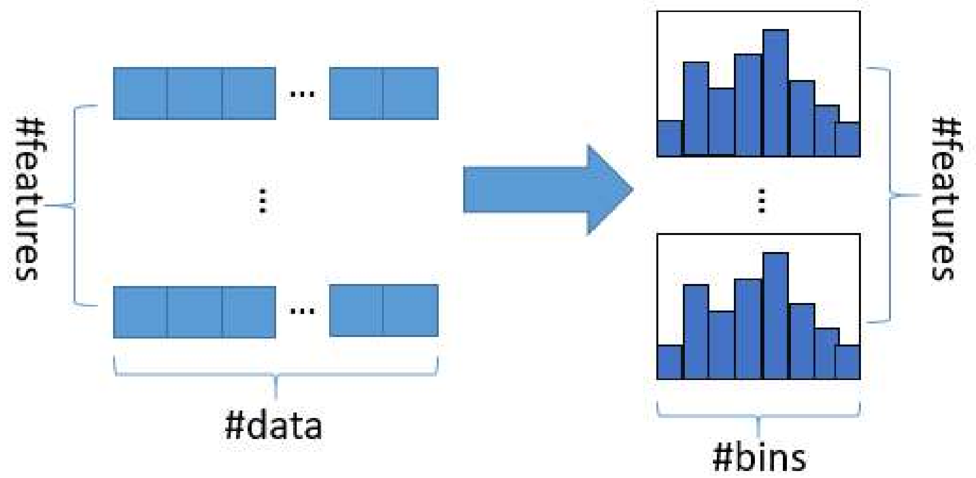
1）压缩了数据的数量；

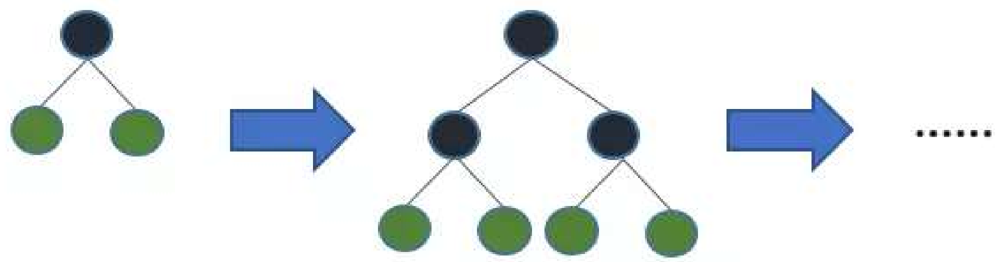
2）压缩了数据的维度；

3）降低训练数据的量。

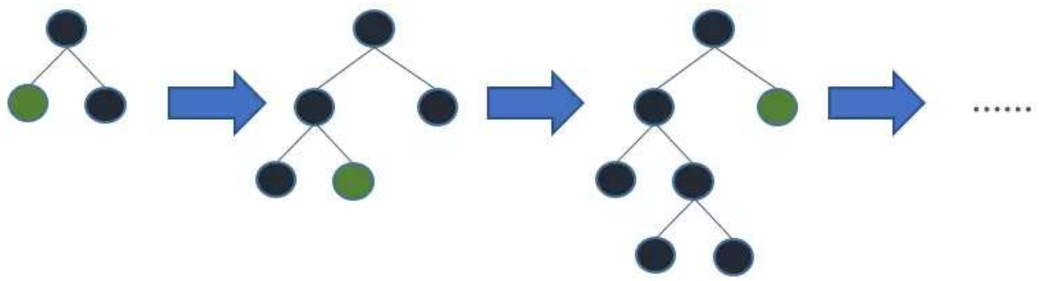
特点

特点	备注
Gradient-based One-Side Sampling（GOSS）	保留梯度较大数据：将分裂的特征按绝对值大小降序排序，XGB:保留排序后结果，LGB不保留取绝对值最大a *100%,剩余小梯度随机选取b*100%,且(1-a)/b。只使用(a+b)%部分数据计算收益
Exclusive Feature Bundling	特征融合绑定降低特征数量：1）图着色：每个特征有个图G定点，用边连接不相互独立的特征，边权重为两特征总冲突值，如着色一样，变为一个bundle;2)对非零值的数量降序排序（进行步骤1，判断是否新建bundle,特征值中加入偏置常量解决捆绑互斥特征，也就是他们很少同时取非零值
Histogram-based Algorithm	连续变量离散化：连续的特征映射到离散的buckets中，组成一个个的bins
Leaf-wise	深度限制的叶子生长：只需达到设置树的叶子数不加深度生长了（速度快之一）





Level-wise tree growth



Leaf-wise tree growth

1.4 xgboost和lightgbm的简单实现

1.5作业

1. 自己实现原始和sklearn接口 xgboost,lightgbm训练
2. 注册国内国外竞赛网站