出现背景：（Facebook提出）

FFM只能完成二阶的特征交叉，GBDT+LR 可以处理更高维度特征交叉所带来的组合爆炸和计算复杂度过高问题。LR模型不能进行特征交叉。

大致思路:

利用梯度提升树（GBDT）来自动特征筛选和组合生成新的离散特征向量，再把特征向量当做LR模型输入。

GBDT模型：

提升树是迭代多棵回归树来共同决策。当采用平方误差损失函数时，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

1.预测方式：将所有的子树结果加起来。

$$D(x) = d_{tree1}(x) + d_{tree2}(x) + \dots.$$

2.构建方式：

通过逐一生成子树方式生成树林，主要利用了样本标签值与当前树林预测值之间的残差构建新的子树。

假设现在有三棵树：

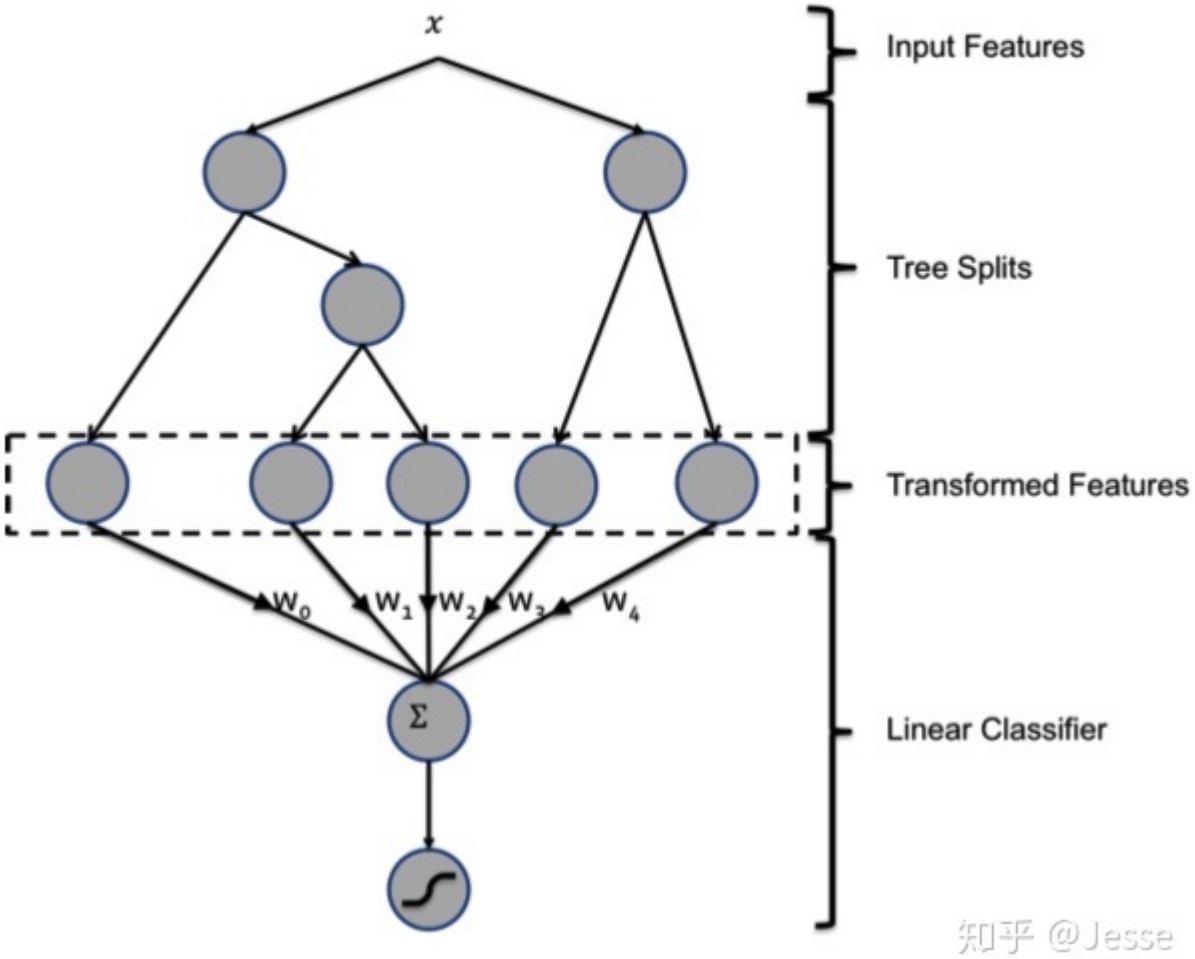
$$D(x) = d_{tree1}(x) + d_{tree2}(x) + d_{tree3}$$

期望构建第四棵，那么有：

$$D(x) + d_{tree4} = f(x)$$

根据残差R(x)为目标有：

$$R(x) = f(x) - D(x)$$



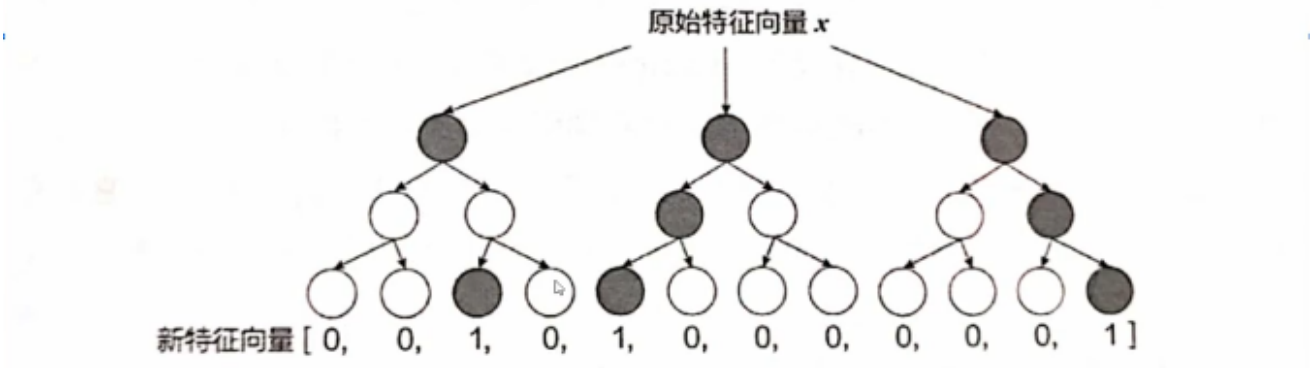
GBDT中可以把树的生成过程理解为自动进行多维度特征组合和特征筛选的过程。

1.回归树中，每个节点的分裂是一个自然的【特征选择】的过程。

2.多层节点的结构则对特征进行了有效的组合（性别，年龄等）。

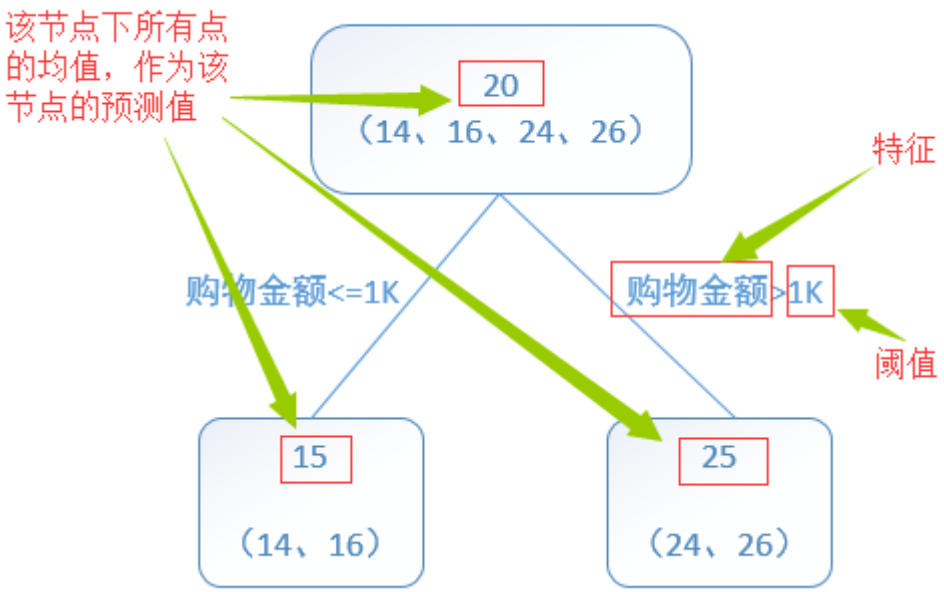
3.不同的样本，会有不同的特征选择以及特征组合。

举例来说，如图 2-17 所示，GBDT 由三棵子树构成，每棵子树有 4 个叶子节点，输入一个训练样本后，其先后落入“子树 1”的第 3 个叶节点中，那么特征向量就是[0,0,1,0]，“子树 2”的第 1 个叶节点，特征向量为[1,0,0,0]，“子树 3”的第 4 个叶节点，特征向量为[0,0,0,1]，最后连接所有特征向量，形成最终的特征向量[0,0,1,0,1,0,0,0,0,0,0,0,1]。

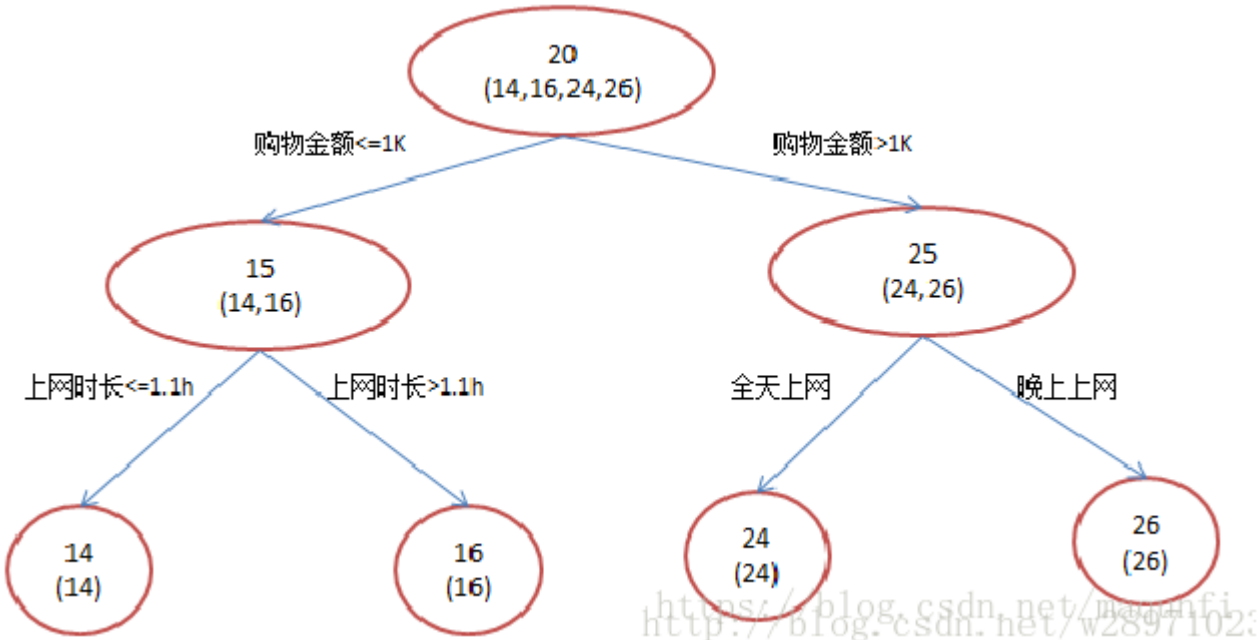


补充：

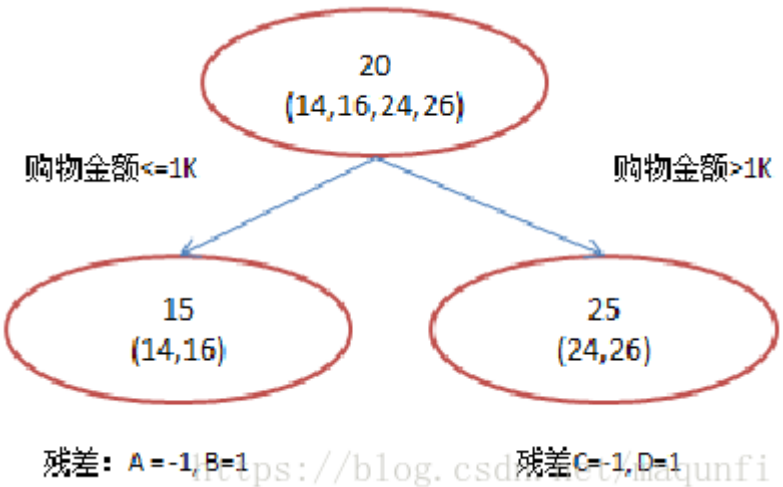
1.回归树总体流程类似于分类树，区别在于，回归树的每一个节点都会得一个预测值，以年龄为例，该预测值等于属于这个节点的所有人年龄的平均值。分枝时穷举每一个feature的每个阈值找最好的分割点，但衡量最好的标准不再是最大熵，而是最小化平方误差。也就是被预测出错的人数越多，错的越离谱，平方误差就越大，通过最小化平方误差能够找到最可靠的分枝依据。分枝直到每个叶子节点上人的年龄都唯一或者达到预设的终止条件(如叶子个数上限)，若最终叶子节点上人的年龄不唯一，则以该节点上所有人的平均年龄做为该叶子节点的预测年龄。



1.假定训练集中有四个人A、B、C、D，年龄分别是14,16,24,26（目标值），划分特征有购物金额<=1，在网时长，经常到百度知道提问，是否上网大于1.1h,是够全天上网这四个特征 使用这些特征构造GBDT模型，去预测对应的年龄。 假如我们使用一个回归树进行训练可以得到下面的树结构。



2.当我们使用GBDT的方法，由于A,B年龄较为相近，C,D年龄较为相近，他们被分为两拨，每拨用平均年龄作为预测值。此时计算残差（残差的意思就是： A的预测值 + A的残差 = A的实际值），所以A的残差就是16-15=1（***注意，A的预测值是指前面所有树累加的和，这里前面只有一棵树所以直接是15，如果还有树则需要都累加起来作为A的预测值***）。进而得到A,B,C,D的残差分别为-1,1，-1,1，这样第一次得到的回归树如下。



3.然后我们拿残差替代A,B,C,D的原值，到第二棵树去学习，如果我们的预测值和它们的残差相等，则只需把第二棵树的结论累加到第一棵树上就能得到真实年龄了。这里的数据显然是我可以做的，第二棵树只有两个值1和-1，直接分成两个节点。此时所有人的残差都是0，即每个人都得到了真实的预测值，从而可以构造出第二棵回归树。

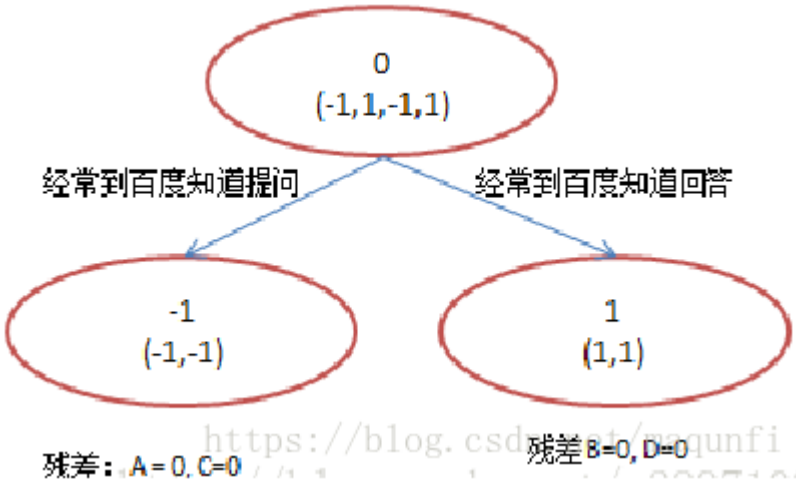
4.当此时把两颗进行结合（没有计算权重），就能根据特征得到每个人对应的预测年龄。

```
* 购物较少，经常到百度知道提问；预测年龄A = 15 - 1 = 14

* 购物较少，经常到百度知道回答；预测年龄B = 15 + 1 = 16

* 购物较多，经常到百度知道提问；预测年龄C = 25 - 1 = 24

* 购物较多，经常到百度知道回答；预测年龄D = 25 + 1 = 26
```



算法 5.5（最小二乘回归树生成算法）

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

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \tag{5.21}$$

遍历变量 j，对固定的切分变量 j 扫描切分点 s，选择使式 (5.21) 达到最小值的对 (j,s).

（2）用选定的对 (j,s) 划分区域并决定相应的输出值：

$$R_1(j,s)=\{x\,|\,x^{(j)}\leqslant s\},\quad R_2(j,s)=\{x\,|\,x^{(j)}>s\}$$

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

（3）继续对两个子区域调用步骤 (1)，(2)，直至满足停止条件.

（4）将输入空间划分为 M 个区域 $R_1,R_2,\cdots,R_M$，生成决策树：

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

■

GBDT 多分类举例说明

样本编号	花萼长度(cm)	花萼宽度(cm)	花瓣长度(cm)	花瓣宽度	花的种类
1	5.1	3.5	1.4	0.2	山鸢尾
2	4.9	3.0	1.4	0.2	山鸢尾
3	7.0	3.2	4.7	1.4	杂色鸢尾
4	6.4	3.2	4.5	1.5	杂色鸢尾
5	6.3	3.3	6.0	2.5	维吉尼亚鸢尾
6	5.8	2.7	5.1	1.9	维吉尼亚鸢尾

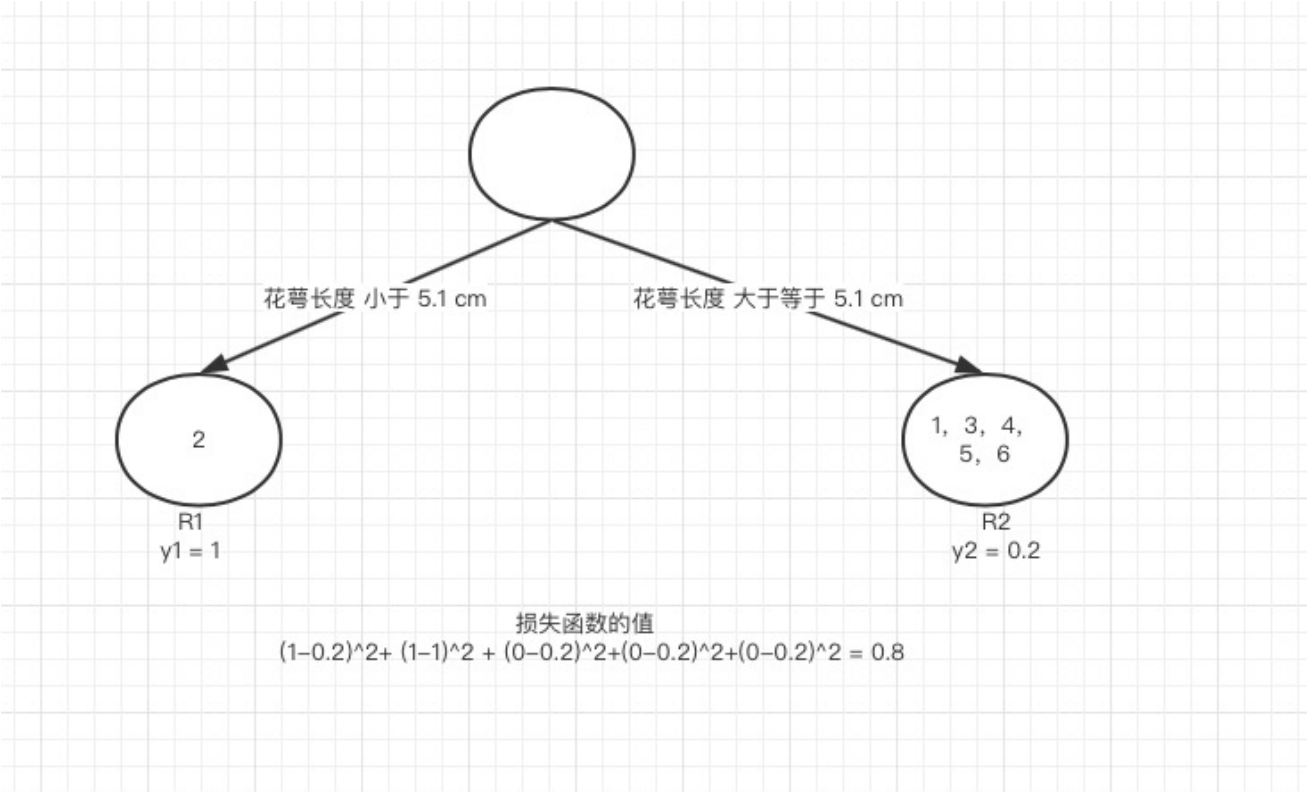
这是一个有6个样本的三分类问题。我们需要根据这个花的花萼长度，花萼宽度，花瓣长度，花瓣宽度来判断这个花属于山鸢尾，杂色鸢尾，还是维吉尼亚鸢尾。具体应用到gbdt多分类算法上面。我们用一个三维向量来标志样本的label。[1,0,0] 表示样本属于山鸢尾，[0,1,0] 表示样本属于杂色鸢尾，[0,0,1] 表示属于维吉尼亚鸢尾。

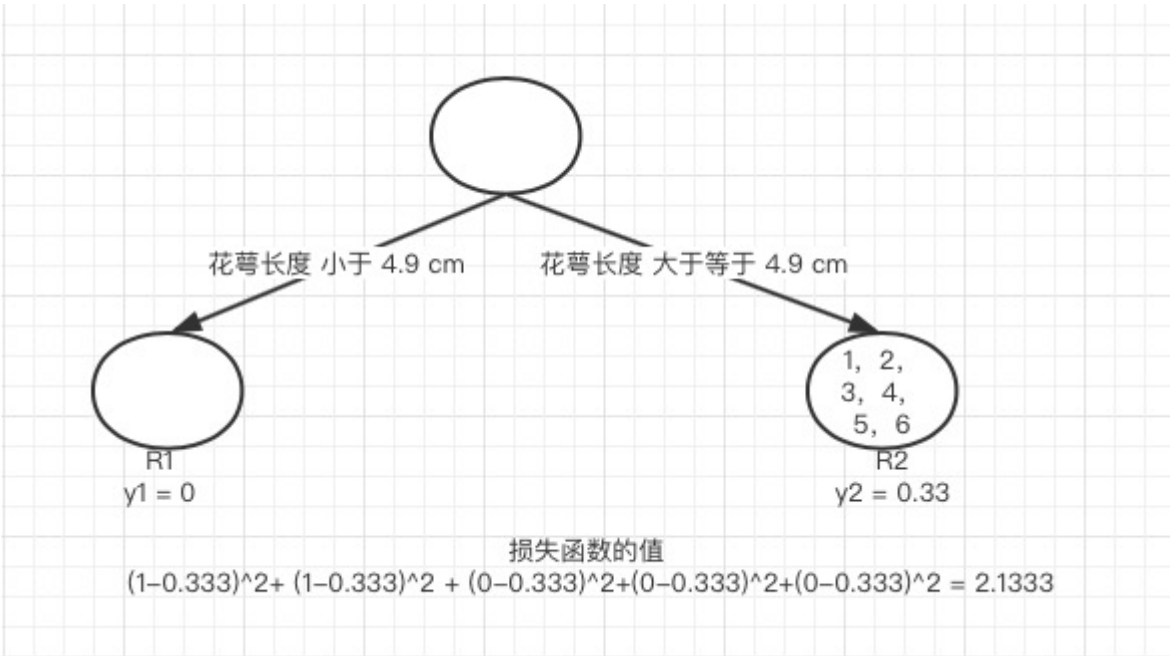
gbdt 的多分类是针对每个类都独立训练一个 CART Tree。所以这里，我们将针对山鸢尾类别训练一个 CART Tree 1。杂色鸢尾训练一个 CART Tree 2 。维吉尼亚鸢尾训练一个CART Tree 3，这三个树相互独立。

我们以样本 1 为例。针对 CART Tree1 的训练样本是[5.1,3.5,1.4,0.2][5.1,3.5,1.4,0.2]，label 是 1，最终输入到模型当中的为[5.1,3.5,1.4,0.2,1][5.1,3.5,1.4,0.2,1]。针对 CART Tree2 的训练样本也是[5.1,3.5,1.4,0.2][5.1,3.5,1.4,0.2],但是label 为 0,最终输入模型的为[5.1,3.5,1.4,0.2,0][5.1,3.5,1.4,0.2,0]. 针对 CART Tree 3的训练样本也是[5.1,3.5,1.4,0.2][5.1,3.5,1.4,0.2],label 也为0,最终输入模型当中的为[5.1,3.5,1.4,0.2,0][5.1,3.5,1.4,0.2,0].

下面我们来看 CART Tree1 是如何生成的，其他树 CART Tree2 , CART Tree 3的生成方式是一样的。CART Tree的生成过程是从这四个特征中找一个特征做为CART Tree1 的节点。比如花萼长度做为节点。6个样本当中花萼长度大于5.1 cm的就是 A类，小于等于 5.1 cm 的是B类。生成的过程其实非常简单，问题 1.是哪个特征最合适？ 2.是这个特征的什么特征值作为切分点？ 即使我们已经确定了花萼长度做为节点。花萼长度本身也有很多值。在这里我们的方式是遍历所有的可能性，找到一个最好的特征和它对应的最优特征值可以让当前式子的值最小。

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





LS-PLM模型（MLR）

针对问题：LR表达能力差（阿里巴巴主流推荐模型）

解决思路：

- 通过聚类，构造【带权】LR
- 先对样本进行分片（聚类），再在样本分片中应用逻辑回归进行CTR评估

$$f(x) = \sum_{i=1}^{m} \pi_i(x) \cdot \eta_i(x) = \sum_{i=1}^{m} \frac{e^{\mu_i \cdot x}}{\sum_{j=1}^{m} e^{\mu_j \cdot x}} \cdot \frac{1}{1 + e^{-w_i \cdot x}}$$

聚类函数
这里用的是softmax

逻辑回归

其中m代表分片数，当m=1时退化为LR模型，m越大拟合能力越强规模也就越大，阿里给出的m的经验值为12.

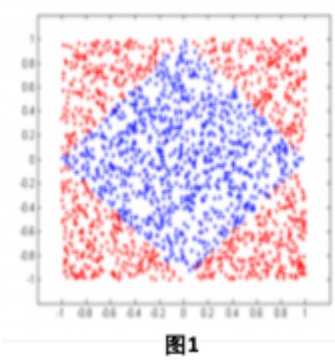


图1

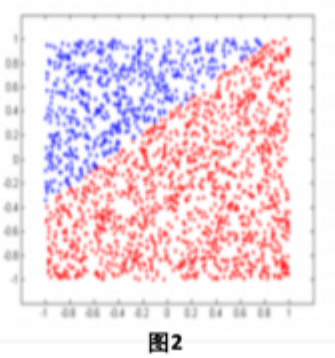


图2

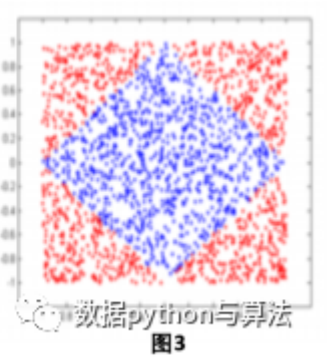
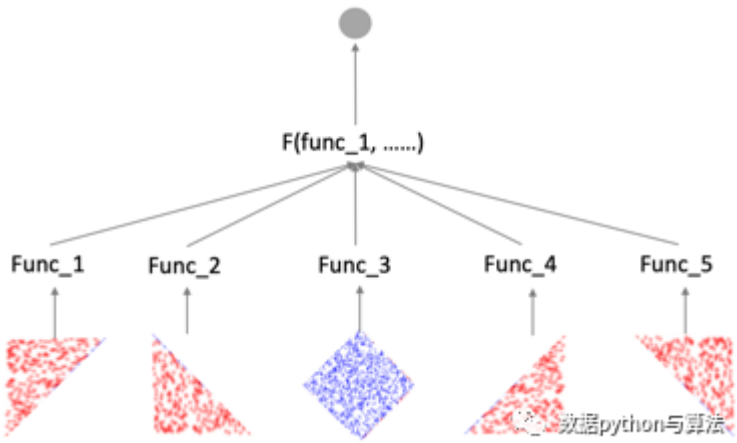


图3



目标函数

目标函数如公式（2）所示

$$\arg\min_{\Theta} f(\Theta) = \text{loss}(\Theta) + \lambda\|\Theta\|_{2,1} + \beta\|\Theta\|_1 \qquad (2)$$

$\text{loss}(\Theta)$根据不同场景下不同，比如二分类用交叉熵作为损失函数：

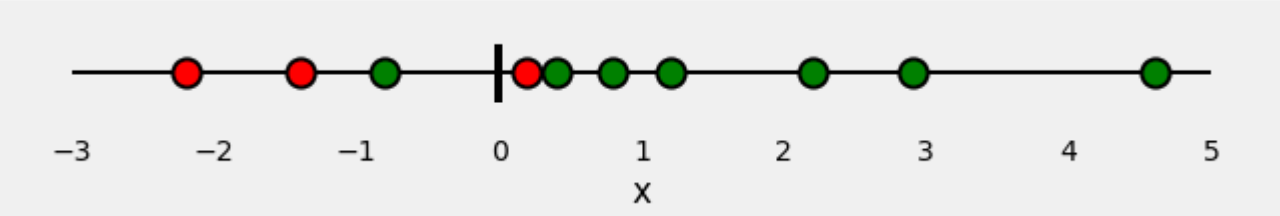
$$\text{loss}(\Theta) = -\sum_{t=1}^{n} \left[y_t \log\left(p\left(y_t = 1 | x_t, \Theta\right)\right) + \left(1 - y_t\right) \log\left(p\left(y_t = 0 | x_t, \Theta\right)\right) \right]$$

后面为正则项，其中
$\|\Theta\|_1 = \sum_{ij} |\theta_{ij}|$是对每个参数的L1正则，保证所有参数的稀疏性；
$\|\Theta\|_{2,1} = \sum_{i=1}^{d} \sqrt{\sum_{j=1}^{2m} \theta_{ij}^2}$是对L2正则的L1正则，根号里面是对某个特征的$2m$个参数的L2正则，外面是L1,这样是为了保证特征的稀疏性，做feature selection。
正则部分相当于对整体参数和按照特征分组的参数分别做了正则，既在最细粒度上筛选参数也在较粗的粒度上筛选特征。

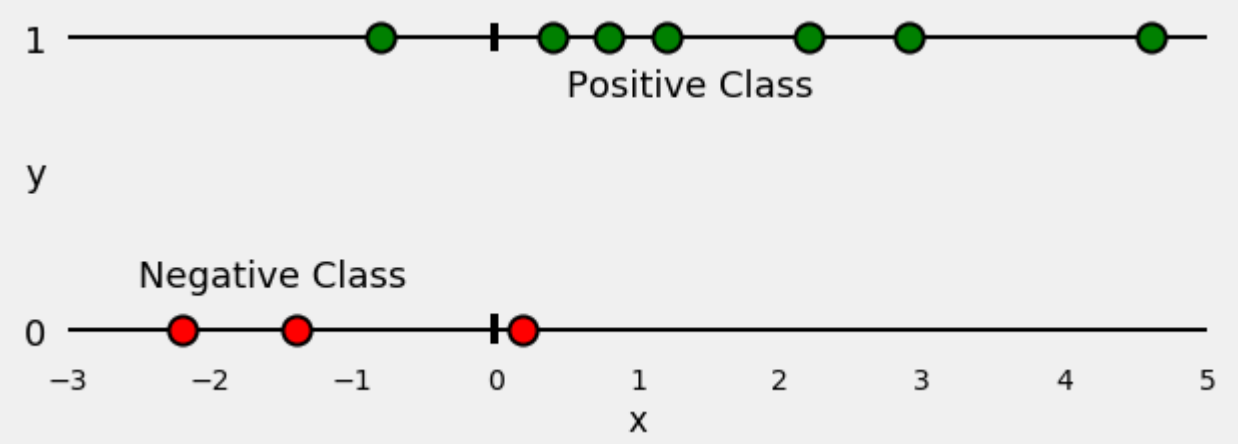
补充：

二元交叉熵

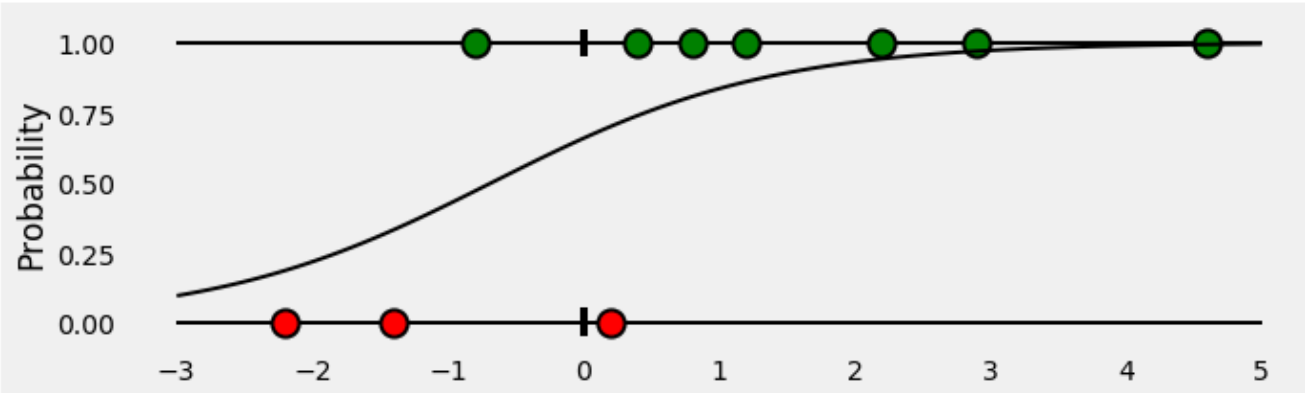
$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$



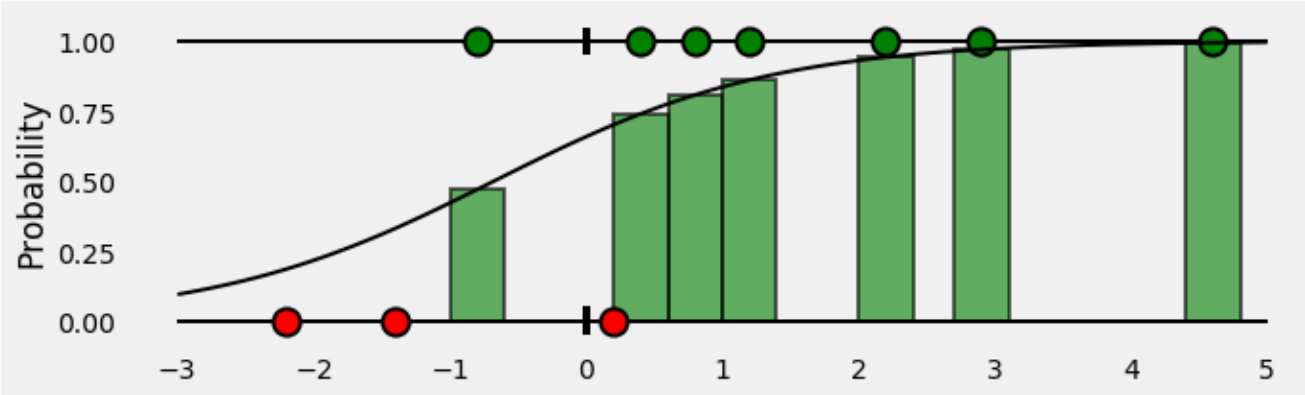
y是标签（1是绿色的，0是红色的），p(y)是所有的N个点预测是绿色的概率。



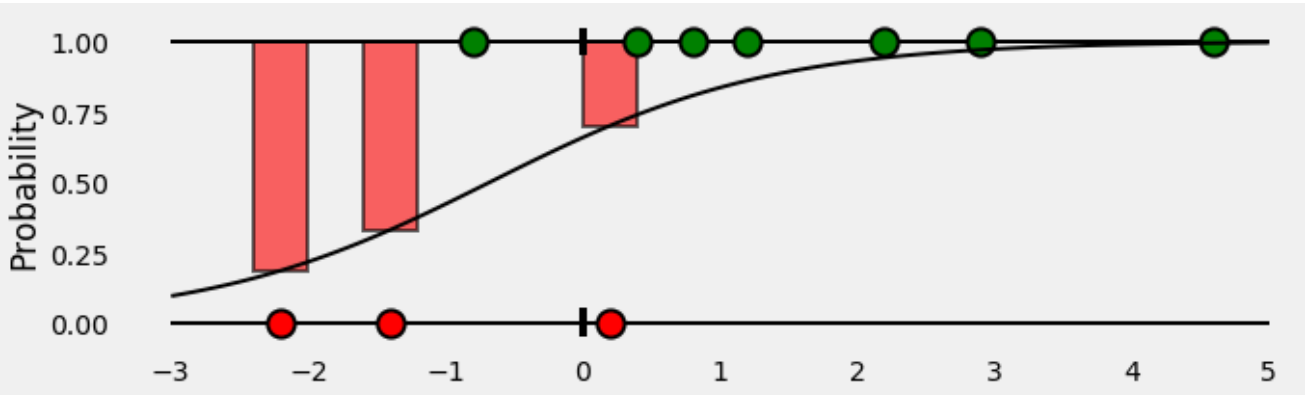
现在，我们来训练逻辑回归模型来分类我们的点。这个回归的拟合是一个sigmoid的曲线，表示了给定的x是绿色的概率。就像这样：

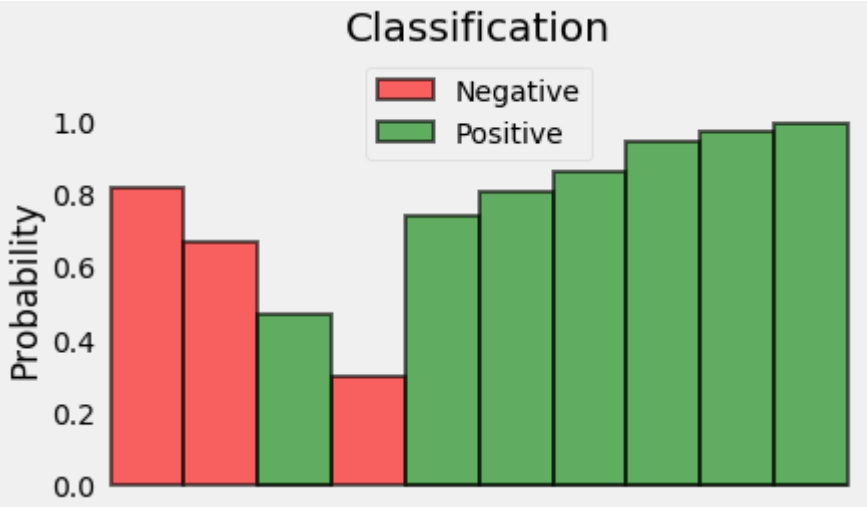


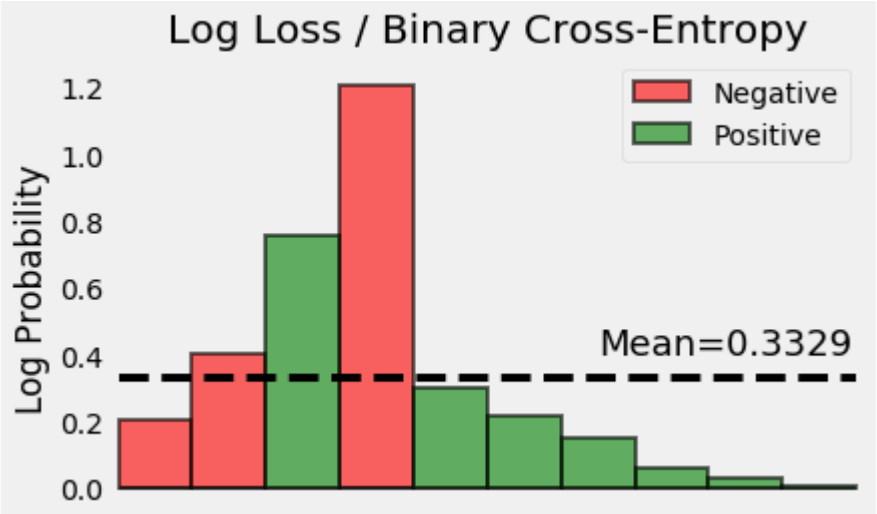
对于所有的属于正样本的点（绿色），我们的分类器给出的预测概率是什么？就是sigmoid曲线下面的绿色的条，x的坐标代表了这个点。



到现在为止，一切都好！那么负样本的点呢？记住，sigmoid曲线之下的绿条表示的该点是绿色的概率。那么，给定的点是红色的概率是多少呢？当然就是sigmoid曲线上面红色条



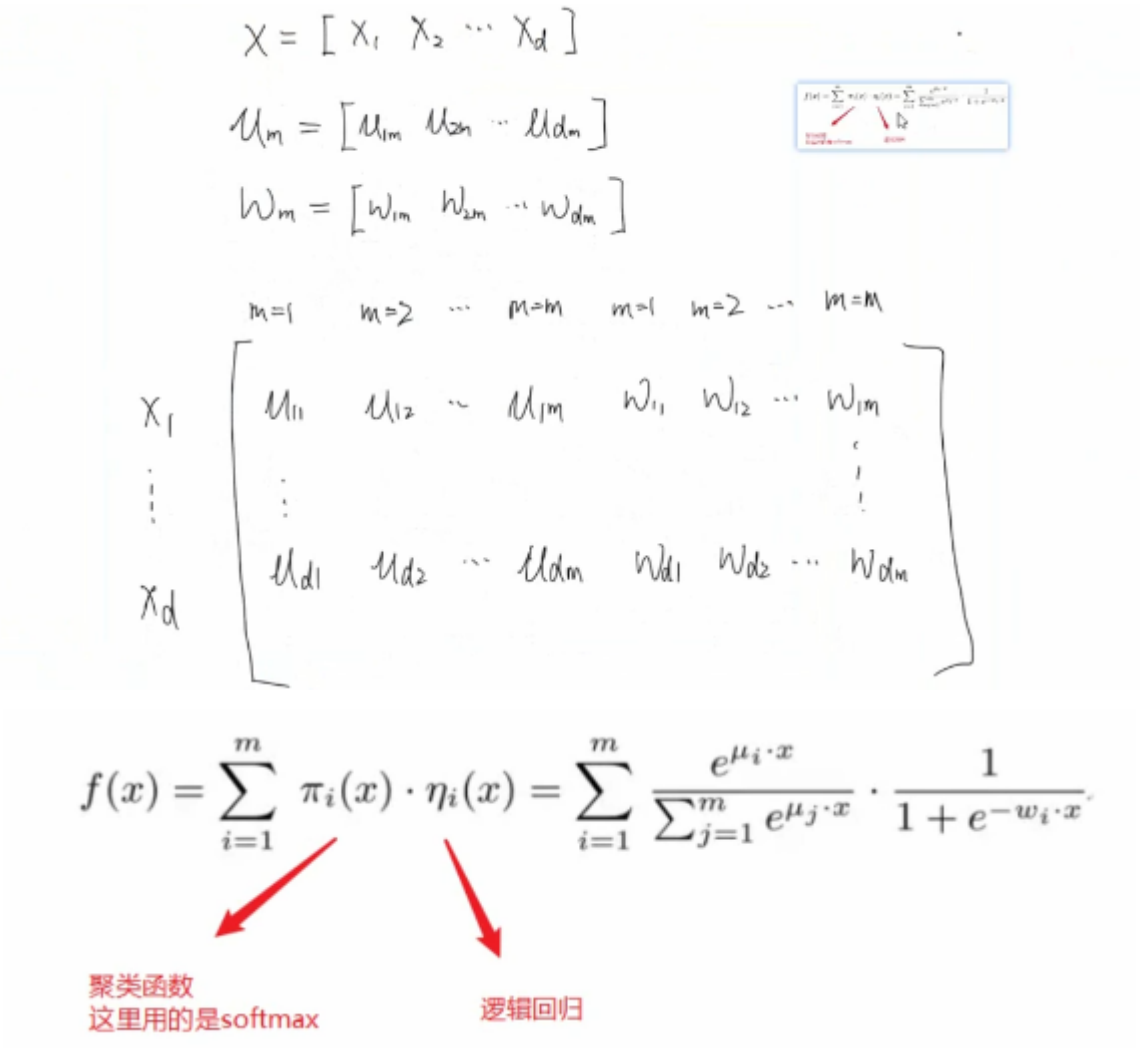


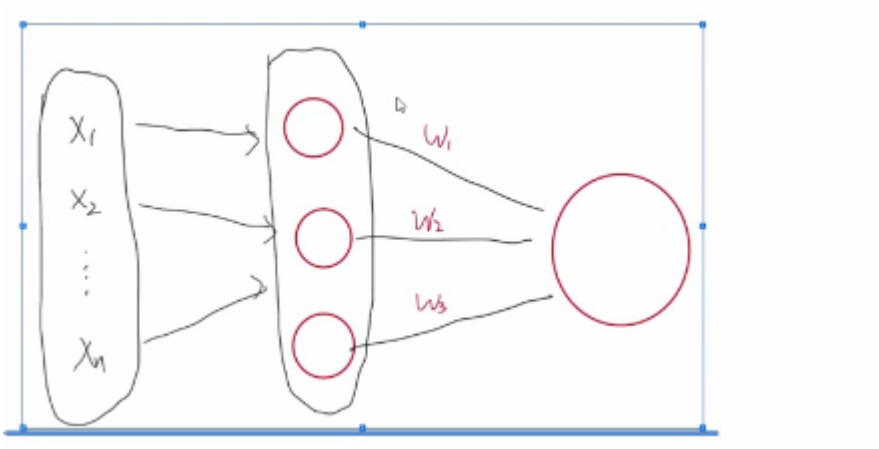


L1L2L2,1正则：

L1正则化和L2正则化的说明如下：

- L1正则化是指权值向量ww中各个元素的**绝对值之和**
- L2正则化是指权值向量ww中各个元素的**平方和然后再求平方根**





MLR可以看做是加入了注意力的三层神经网络，输入层是样本的特征向量，中间层是m个神经元组成的隐层，其中m是分片的个数，最后一层是单个神经元的输出层。

问题：

如何将多个回归树建成一个树（有无顺序要求）？