规则篇

《100天成为风控专家》

规则生成(3)：决策树(含实操)

出品：东哥起飞

解锁风控课程

关注我的公众号

目录

扫码加我微信

扫码加我微信

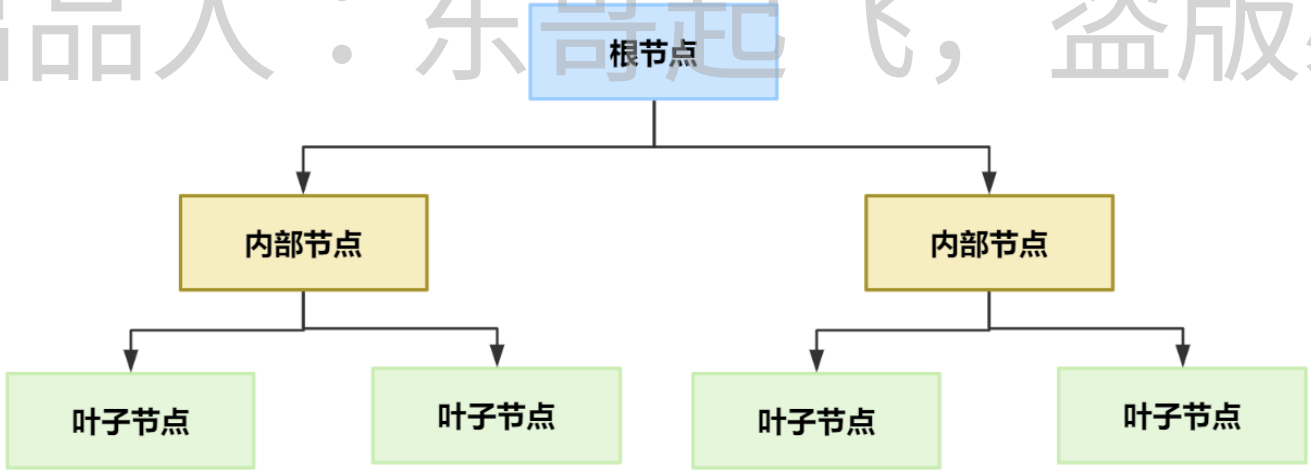
一、决策树的概念

1.1. 什么是决策树?

决策树是一个if-then规则的集合，自顶向下按照某种度量标准不断地进行分裂，形状上如同一个树形的结构。

在树结构中（如下图），包括两种类型的节点：**分裂节点、末端节点**。分裂节点按照特征变量阈值的判断会向下分裂；而末端节点，也叫**叶子节点**是决策树的末梢终点，不会继续分裂。

分裂节点中又分**根节点、内部分裂点**，本质上是一样的，二者区别是根节点是决策树的起始节点，而内部分裂点在决策树内部。

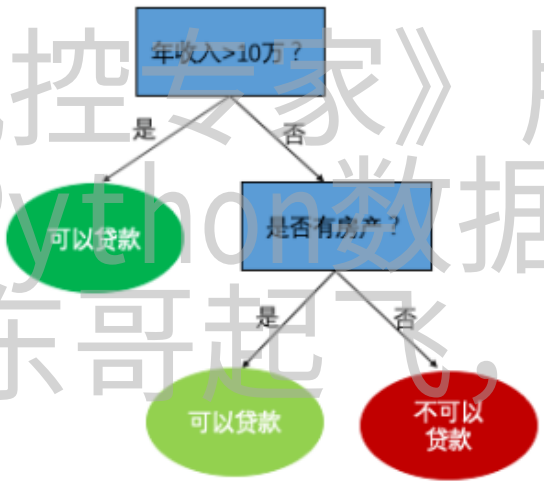


1.2. 决策树的生成过程

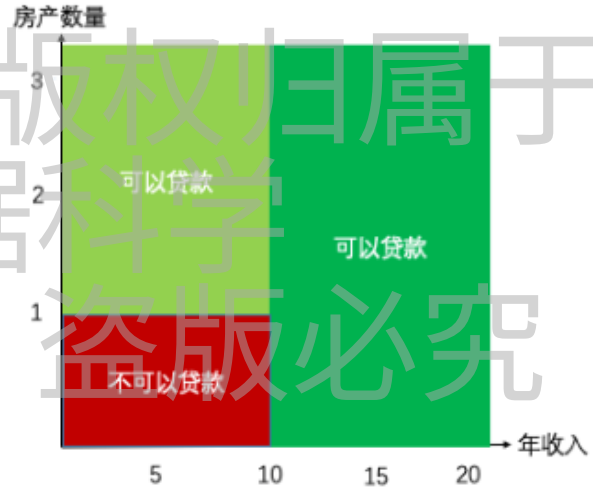
那么决策树是如何生成的呢?

简单来说，决策树从根节点开始，在每个分裂节点均会按照“一定的度量标准”从特征变量池中筛选出最合适的变量以及变量对应的阈值，并不断地向下分裂，直到达到某种条件后停止分裂，最终输出叶子节点的数值或类别结果。

右侧示例就是决策树的一般生成过程，相当于通过特征变量的选择对样本的特征空间做非线性地划分，因此我们说决策树也是一种“**非线性模型**”。



if then决策规则的层次结构



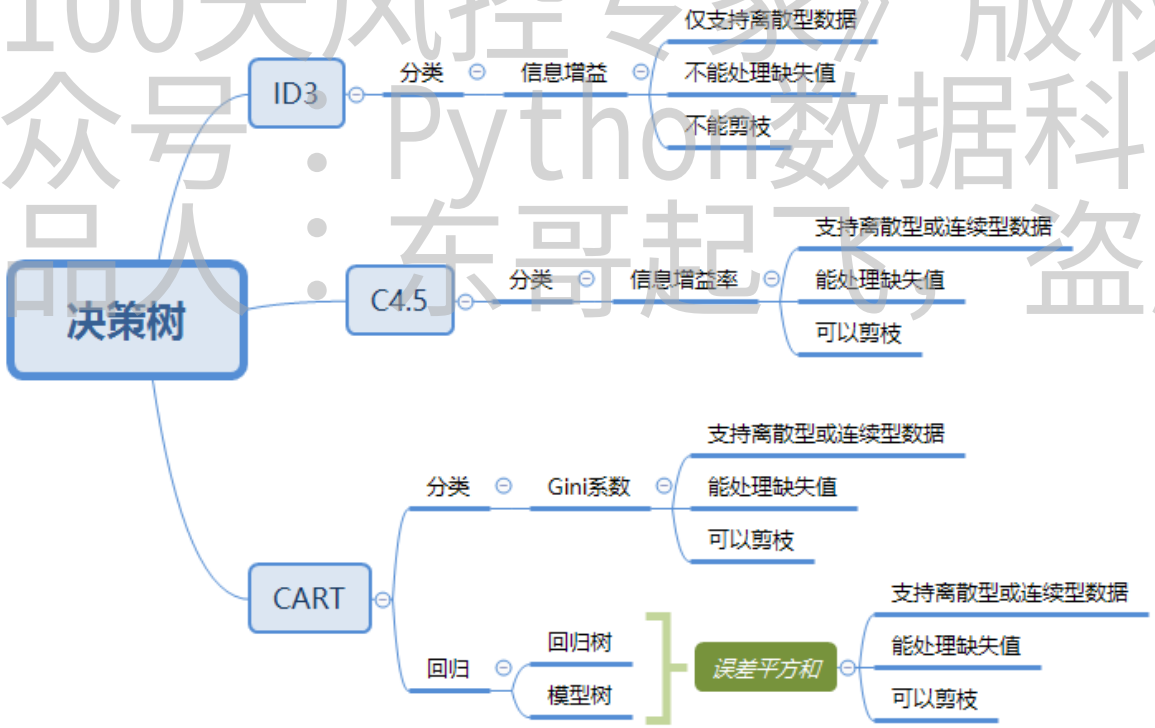
特征空间的不相交子区域划分

扫码加我微信

二、决策树算法

2.1. 算法分类

决策树有三种算法：ID3、C4.5、CART，其中ID3和C4.5用于分类，而CART既可以分类，也可以回，并且是二叉树计算更快。下面我们以最常用的**CART算法**来举例说明，因为它也是后面模型篇各种集成树模型的基础，是比较重要的。



2.2. CART分类树—基尼系数

CART是 "Classification and Regression Trees" 的缩写，意思是 "**分类回归树**"。从它的名字上就不难理解了，CART算法是既可以用于分类的，也可以用于回归的。

基尼系数(Gini)

CART分类树算法使用“基尼系数(Gini)”选择特征，**基尼系数代表了模型的不纯度，基尼系数越小，不纯度越低，特征越好。**公式如下：

$$Gini(D) = \sum_{k=1}^{K} p(x_k) * (1 - p(x_k))$$

$$= 1 - \sum_{k=1}^{K} p(x_k)^2$$

当n=2时

$$Gini(p) = 2p(1-p)$$

这个公式中，p(x_k)表示分类x_k出现的概率，K是分类的数目。比如在信贷风控中，我们的分类是好坏客户只有两类，基尼指数就等于2p(1-p)。

2.2. CART分类树—基尼系数

对于给定的样本集合D，其基尼指数定义为：

$$Gini(D) = \sum_{k=1}^{K} \frac{|C_k|}{|D|}(1 - \frac{|C_k|}{|D|})$$

$$= 1 - \sum_{k=1}^{K} (\frac{|C_k|}{|D|})^2$$

$$Gini(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} Gini(D_i)$$

其中， k 代表类别，Ck是D中属于第k类的样本子集。

当 CART 为二分类时，则在特征A的条件下，集合D的基尼指数定义为：

$$Gini(D|A) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

2.2. CART分类树—基尼系数

在信贷场景中我们想区分好坏客户，那么目标变量有两类：好客户和坏客户。我们举两个极端的情况说明。

1）如果分裂后的节点中好坏客户各占50%，此时基尼指数为0.5，不纯度达到最大值，说明完全没有任何区分效果。

$$
\begin{aligned}
Gini(p) &= 2p(1-p) \\
&= 2 * \frac{1}{2} * (1-\frac{1}{2}) \\
&= 0.5
\end{aligned}
$$

2）如果分裂后的节点中好客户占比100%，坏客户占比0%或者反过来，基尼系数为0，不纯度达到最小值，此时区分效果最强，完全是好或者坏客户，一边倒。

$$
\begin{aligned}
Gini(p) &= 2p(1-p) \\
&= 2 * 1 * (1-1) \\
&= 0
\end{aligned}
$$

基尼指数反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率。因此，Gini(D)越小，则数据集D的纯度越高。

2.3. CART分类树—变量二分法

对于连续型变量，假如变量a有连续值m个，从小到大排列。m个数值就有m-1个切分点，分别使用每个切分点把连续数值离散划分成两类，将分裂前数据集D按照划分点分为D1和D2两个子集，然后计算每个划分点下对应的基尼指数，选择值最小的一个作为最终的变量划分。

Split 1：a_1 | a_2 a_3 a_4 a_5 a_6 a_7 … a_{m-1} a_m

$$\frac{a_1+a_2}{2}$$

$$Gini_1(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2)$$

Split 2：a_1 a_2 | a_3 a_4 a_5 a_6 a_7 … a_{m-1} a_m

$$\frac{a_2+a_3}{2}$$

$$Gini_2(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2)$$

… … …

Split m-1：a_1 a_2 a_3 a_4 a_5 a_6 a_7 … a_{m-1} | a_m

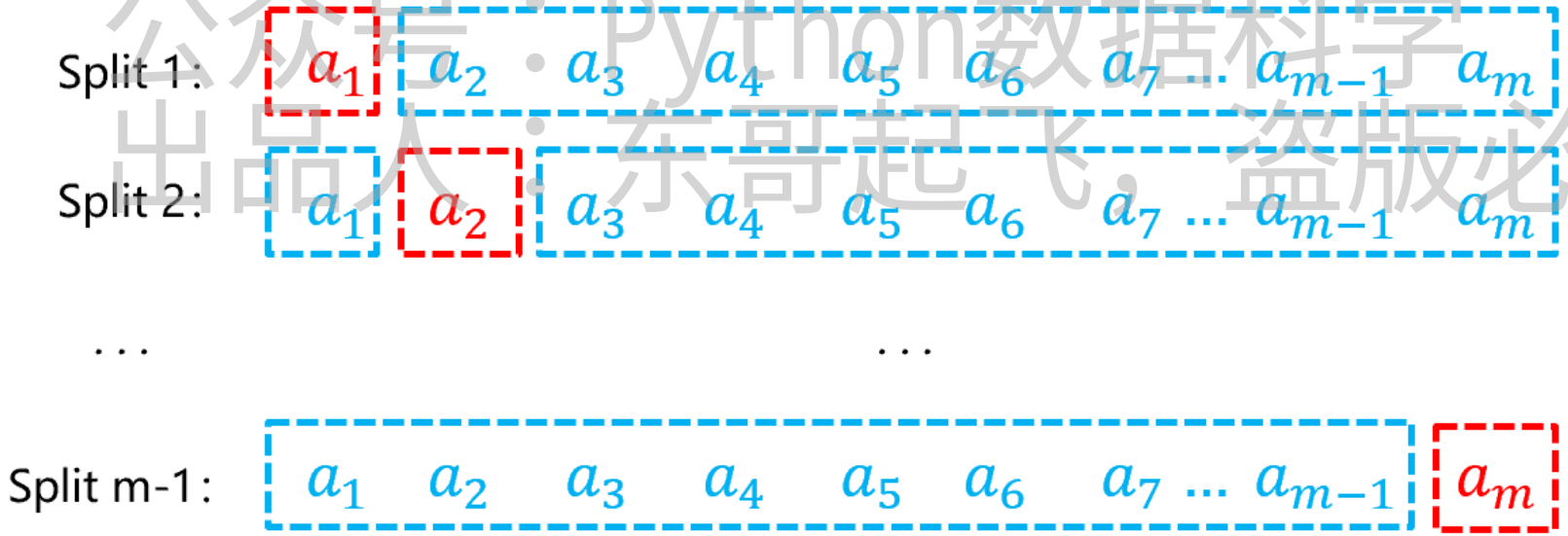
$$\frac{a_{m-1}+a_m}{2}$$

$$Gini_{m-1}(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+\frac{|D_2|}{|D|}Gini(D_2)$$

2.3. CART分类树—变量二分法

对于离散型变量，如果离散值多于两个，CART同样会不停的二分，将其中一个类别作为一类，其余所有类别归为一类。比如下图示例中，离散变量a有m个类别，split1中将a1作为一类，剩余a2-am归为一类，split2中将a2作为一类，其余归为另一类，直到split(m-1)划分都是同理。

与连续型变量处理方式一样，每次划分后分为D1和D2两个子集，然后计算每个划分点下对应的基尼指数，选择值最小的一个作为最终的变量划分。



2.4. CART分类树—递归生成

扫码加我微信

输入：训练集D，基尼系数的阈值，切分的最少样本个数阈值

输出：分类树T

算法从根节点开始，用训练集递归建立CART分类树。

① 对于当前节点的数据集为D，如果样本个数小于阈值或没有特征，则返回决策子树，当前节点停止递归；

② 计算样本集D的基尼系数，如果基尼系数小于阈值，则返回决策树子树，当前节点停止递归；

③ 计算当前节点现有各个特征的各个值的基尼指数；

④ 在计算出来的各个特征的各个值的基尼系数中，选择基尼系数最小的特征A及其对应的取值a作为最优特征和最优切分点。 然后根据最优特征和最优切分点，将本节点的数据集划分成两部分D1和D2，同时生成当前节点的两个子节点，左节点的数据集为D1 ，右节点的数据集为D2；

⑤ 对左右的子节点递归调用1-4步，生成CART分类树；

对生成的CART分类树做预测时，假如测试集里的样本落到了某个叶子节点，而该节点里有多个训练样本。则该测试样本的类别为这个叶子节点里概率最大的类别。

2.5. CART回归树—预测方式

与分类树不同，回归树的预测变量是连续值，比如预测一个人的年龄，又或者预测季度的销售额等等。另外，回归树在**选择特征的度量标准**和**决策树建立后预测的方式**上也存在不同。

① 预测方式

一个回归树对应着输入特征空间的一个划分，以及在划分单元上的输出值。现在假设数据集已被划分，R_1,R_2,...,R_m共m的子集，回归树要求每个划分R_m中都对应一个固定的输出值c_m。

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

这个c_m值其实就是每个子集中所有样本的目标变量y的平均值，并以此c_m作为该子集的预测值。所有分支节点都是如此，叶子节点也不例外。因此，可以知道回归树的预测方式是：**将叶子节点中样本的y均值作为回归的预测值。而分类树的预测方式则是：叶子节点中概率最大的类别作为当前节点的预测类别。**

2.6. CART回归树—残差平方和

② 选择特征的度量标准

CART回归树对于变量类型的处理与分类树一样，连续值与离散值分开对待，并只能生成二叉树。但是CART回归树对于选择变量的度量标准则完全不同。

分类树的特征选择标准使用基尼指数，而回归树则使用**残差平方和(RSS)**，通过**最小化残差平方和作为判断标准**，公式如下：

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_2\in R_2(j,s)}(y_i-c_2)^2\right]$$

yi：样本目标变量的真实值；
R1&R2：被划分的两个子集；
c1&c2：R1&R2子集的样本均值；
j：当前的样本特征；
s：划分点；

上面公式的含义是：**计算所有的“变量和切分点”组合的残差平方和，找到一组(变量j，切分点s)，以分别最小化左子树和右子树的残差平方和，并在此基础上再次最小化二者之和。**

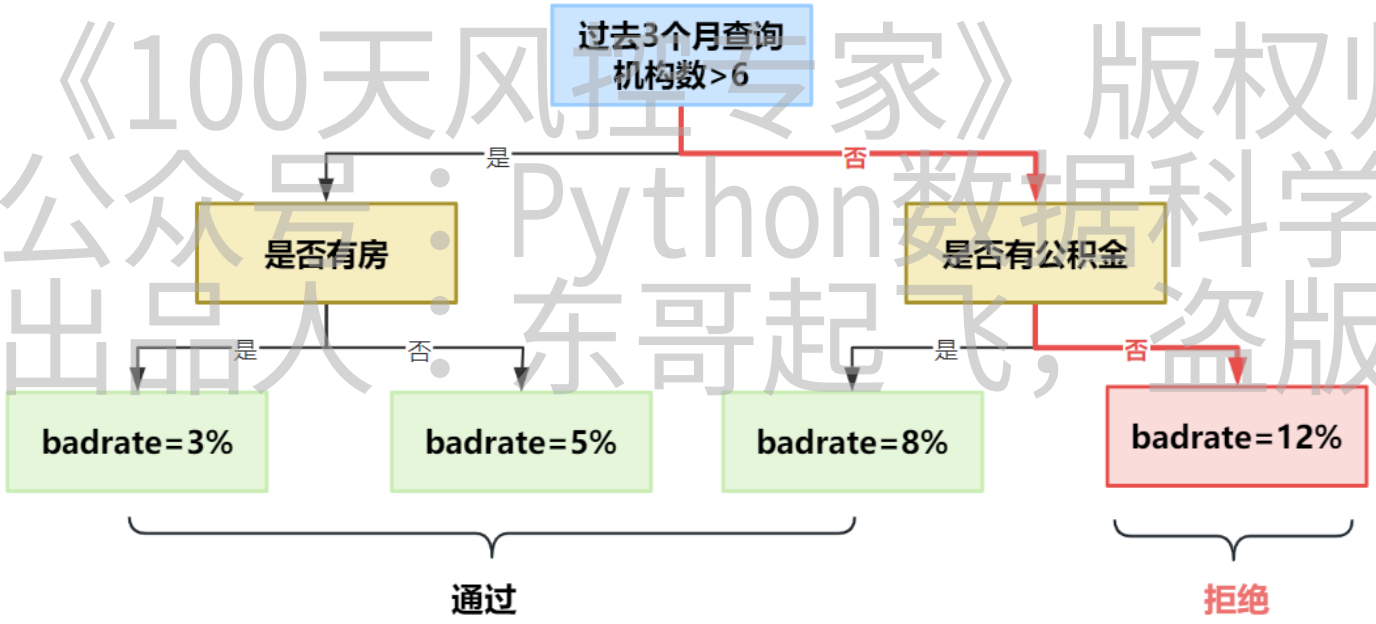
其实，回归树也有分类的思想。所谓 “物以类聚”，相同类之间的目标变量值才会更接近，方差值也就会更小。

扫码加我微信

三、决策树生成规则

3.1. 决策树生成规则过程

以信贷风控场景为例，假如样本的整体坏账率badrate为6%，有征信、公积金、房屋资产等维度的数据，现在基于以上所有变量和目标变量生成一个决策树，树深为2，具体如下。



规则制定：（过于3个月查询机构数>6）且（是否有公积金=否），触发则拒绝，反之通过。

3.2. 决策树规则注意事项

① 规则的复杂度

随着树深度不断增加，对客群的划分更加细致精准，但也容易出现两个问题，一是容易过拟合，在训练样本上效果好，但在时间外OOT样本上效果差；二是树深度过深，会导致生成的规则过于复杂，不利于上线后的监控和运营维护。比如，当规则出现异常时，需要定位变量的原因，对于一条包含很多个变量的规则，排查难度会增加。**在信贷风控中，一条规则包含的变量数不超过3个。**

② 规则评估指标

与单变量、交叉表的评估指标相同，综合考虑精准率、召回率、命中率等指标，以此评估规则的可用性。

③ 规则可解释性

通过决策树生成的规则也需要满足业务的可解释性。

扫码加我微信

四、Python代码案例实操

4.1. Sklearn.tree的API方法

Sklearn中有两个决策树API方法，分别是：

① **tree.DecisionTreeClassifier**：CART分类树

② **tree.DecisionTreeRegressor**：CART回归树

要注意的是，Sklearn没有对ID3和C4.5算法的实现，就只有CART算法，并且是调优过的。下面是官方文档的说明。

https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

1.10.6. Tree algorithms: ID3, C4.5, C5.0 and CART

What are all the various decision tree algorithms and how do they differ from each other? Which one is implemented in scikit-learn?

▶ **Various decision tree algorithms**

scikit-learn uses an optimized version of the CART algorithm; however, the scikit-learn implementation does not support categorical variables for now.

4.2. Sklearn.tree的API参数

DecisionTreeClassifier参数如下：

1. **Criterion:** 特征选择的度量标准，可以选择 "gini" 或 "entropy"。默认情况下使用 "gini"，是CART算法的标准。
2. **Splitter:** 用于指定节点分裂的方式，可以选择 "best" 或 "random"。默认情况下是 "best"，即在所有可能的划分点中选择最优的划分点。"random" 则是随机选择局部最优的划分点。
3. **Max_Depth:** 用于限制决策树的深度，如果没有设置，则表示没有限制。这是一个重要的参数，因为它可以帮助防止决策树变得过于复杂，从而避免过拟合。
4. **Min_Samples_Split:** 指定节点分裂所需的最低样本数。如果某个节点的样本数量少于这个值，则不会进行分裂。
5. **Min_Samples_Leaf:** 指定叶子节点所需的最低样本数。如果某个叶子节点的样本数量少于这个值，会与它的兄弟节点合并。

4.2. Sklearn.tree的API参数

6. **Max_Features:** 用于控制每次寻找最优划分时考虑的特征数量的上限。这有助于减少决策树的复杂性，同时保持模型的准确性。

7. **Random_State:** 随机种子，用于确保每次训练的结果一致性。如果没有设置随机种子，则每次训练可能会得到不同的结果。

8. **Class_Weight:** 用于指定不同类别的权重，以平衡数据集中的类别不平衡问题。

9. **Min_Impurity_Decrease:** 指定分割后的信息增益阈值，如果小于这个值就不继续分裂。

10. **CCP_Alpha:** 剪枝复杂度惩罚项系数，用于指导剪枝过程。

谢谢

出品人：东哥起飞

解锁风控课程

关注我的公众号

扫码加我微信