## 梯度提升树与特征工程

树模型：贪心的寻找最佳分割点，形成二叉树模型

不足之处：

1. 类别型变量。当类别属于高基数（有很多种feature value）时，树模型很难快速切分。只要中间有一次切分不好，会对后续的所有分割造成影响（可以使用Mean-target均值编码）
2. 特征之间的交叉关系。如果两个特征的交叉存在极强的信息，树模型很难挖掘到
3. 上下文信息。比如波动曲线中，观察下一时刻与上一时刻的差值是否大于阈值

* 类别特征编码

1. LabelEncoder：把类别标签转化为0~n\_classes-1
2. OneHotEncoder：M个不同的类别生成M个不同的列，每一列代表一个类别label
3. Frequency：使用类别出现的频次作为新的类别

df[‘color\_cnt’] = df[‘color’].map(df[‘color’].value\_counts())

1. Target编码：从标签y衍生出新的类别特征
   1. Leave-one-out mean target：编码时不考虑当前样本，用其它相同特征值得样本的均值作新的类别特征

from category\_encoders.leave\_one\_out import LeaveOneOutEncoder

loo = LeaveOneOutEncoder()

loo.fit\_transform(df[‘feat’], y)

* 1. K-fold mean target：将训练集分为K折，在对A折进行编码时，删除K折中的A折，用剩余的数据进行计算。

当c的次数很小时，，大概率出现过拟合现象。调大可缓解

from category\_encoders.target\_encoder import TargetEncoder

from sklearn.model\_selection import KFold

kf = KFold()