DataCastle数据城堡

版本v0.1 | 2018年12月28日星期五

揪出营销活动中的羊毛党

数据挖掘案例

目录

[数据挖掘项目的基本流程图 3](#_Toc533843116)

[一、课题背景和目标 4](#_Toc533843117)

[1.1背景 4](#_Toc533843118)

[1.2课题研究目标 4](#_Toc533843119)

[二、数据介绍 4](#_Toc533843120)

[2.1训练集 4](#_Toc533843121)

[2.2测试集 4](#_Toc533843122)

[2.3字段说明 5](#_Toc533843123)

[三、评估准则 7](#_Toc533843124)

[四、认识数据 9](#_Toc533843125)

[4.1特征字段的类型 9](#_Toc533843126)

[4.2 结合业务的分析 10](#_Toc533843127)

[五、数据探索 10](#_Toc533843128)

[5.1可视化结果和结论 11](#_Toc533843129)

[六、数据预处理 14](#_Toc533843130)

[6.1数据预处理的一般方法 14](#_Toc533843131)

[6.2本次比赛数据的预处理 14](#_Toc533843132)

[七、特征工程 15](#_Toc533843133)

[7.1特征创建（第一次） 15](#_Toc533843134)

[7.2特征选择 15](#_Toc533843135)

[八、模型优化 16](#_Toc533843136)

[8.1调参方法 16](#_Toc533843137)

[8.2课程模型与调参 17](#_Toc533843138)

[8.3 XGBoost算法介绍 17](#_Toc533843139)

[8.3.1 XGBoost的优势 17](#_Toc533843140)

[8.3.2 XGBoost的参数 17](#_Toc533843141)

[九、模型评估 21](#_Toc533843142)

[9.1模型评估记录（第一次） 21](#_Toc533843143)

[十、重复七、八、九步直到模型效果无明显提升 22](#_Toc533843144)

[10.1特征工程（第二次） 22](#_Toc533843145)

[10.2模型评估的记录（第二次） 24](#_Toc533843146)

[10.3特征工程（第三次） 25](#_Toc533843147)

[10.4模型评估的记录（第三次） 25](#_Toc533843148)

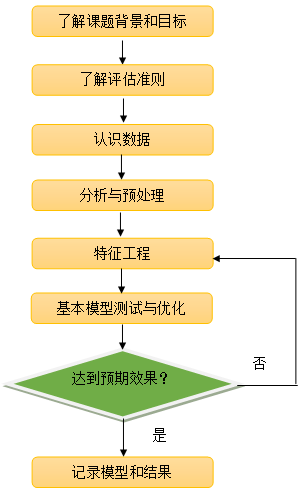
[十一、模型融合 26](#_Toc533843149)

[11.1 Bagging 26](#_Toc533843150)

[11.2 Stacking 27](#_Toc533843151)

[十二、提交 28](#_Toc533843152)

# 数据挖掘项目的基本流程图



# 一、课题背景和目标

## 1.1背景

随着互联网+这一概念的不断发展，电商、出行、外卖等行业近些年也持续发展壮大，越来越多的商家进入这一市场。为了在激烈的竞争中拉取新用户，培养用户的消费习惯，各种类型的营销活动和补贴活动也是层出不穷。在为正常用户带来福利的同时，也催生了一批专注于营销活动的“羊毛党”。目前，羊毛党的行为越发专业化，团伙化和地域化，同套利黑产团伙的斗争，是一场永无止境的攻防战。

机器学习模型是风控系统中实时识别和对抗黑产攻击的有效手段。面对黑产攻击手段快速多变，黑样本数据标签缺失等问题，目前除了LR,RF等耳熟能详的机器学习模型，基于RNN的深度学习模型，无监督学习模型等技术也被应用到同黑产的对抗中。

## 1.2课题研究目标

判断一个用户是否是黑产用户，这是一个典型的二分类问题。

# 二、数据介绍

## 2.1训练集

1. operation\_train\_new.csv为训练集操作详情表单，共1460843条数据；

2. transaction\_train\_new.csv为训练集交易详情表单，共264654条数据；

3. tag\_train\_new.csv为训练集黑白样本标签，共31179 条数据。

## 2.2测试集

1. operation\_round1\_new.csv为测试集操作详情表单，共1769049条数据；

2. transaction\_round1\_new.csv为测试集交易详情表单，共168981条数据。

## 2.3字段说明

**(1)** **operation\_train\_new.csv：**

UID：用户编号

day：操作日期（1， 2， 3，…， 30）

mode：操作类型（例如：修改密码、查询余额等）

success：操作状态（成功/失败）

os：操作系统

version：客户端版本号

device1：设备名称

device2：设备型号

device\_code1：设备唯一标识1，安卓类

device\_code2：设备唯一标识2，安卓类

device\_code3：设备唯一标识3，苹果类

mac1：设备mac地址

mac2：wifi mac地址

ip1：设备ip

ip2：电脑ip

wifi：wifi名称

geo\_code：地理位置

ip1\_sub：ip1前三位

ip2\_sub：ip2前三位

**(2) transaction\_train\_new.csv：**

UID：用户编号

Channel：平台类型

day：交易日期

time：交易时间

trans\_amt：交易金额

amt\_src1：资金类型（例如：余额、银行卡等）

amt\_src2：资金类型，对银行卡做了细分

merchant：商户标识

code1：商户子门店标识

code2：商户设备标识

trans\_type1：交易类型（例如：消费、退款）

trans\_type2：交易类型（例如：线上、线下）

acc\_id1：交易账号

acc\_id2：转出账号

acc\_id3：转入账号

device\_code1：交易设备标识，安卓类

device\_code2：交易设备标识，安卓类

device\_code3：交易设备标识，苹果类

device1：交易设备名称

device2：交易设备型号

mac1：设备mac地址

ip1：设备ip地址

bal：账户余额

geo\_code：地理位置

market\_code：营销活动号

market\_type：营销活动类型

ip1\_sub：ip前三位

**(3) tag\_train\_new：**

UID：用户编号

Tag：是否为黑用户

# 三、评估准则

分类问题评价指标主要是基于混淆矩阵。

在黑产监控中，需要做到尽可能少的误伤和尽可能准确地探测，该课题选择“在FPR较低时的TPR加权平均值”作为平均指标。

给定一个阀值，可根据混淆矩阵计算TPR（覆盖率）和FPR（打扰率）

TPR = TP /（TP + FN） FPR = FP /（FP + TN）

其中，TP、FN、FP、TN分别为真正例、假反例、假正例、真反例。

通过设定不同的阈值，会有一系列TPR和FPR，就可以绘制出ROC曲线

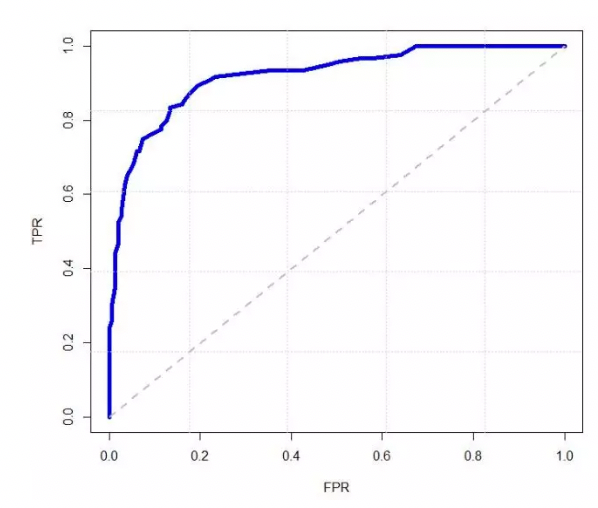


图 3-1 ROC 曲线

这里的评分指标，首先计算了3个覆盖率TPR：

TPR1：FPR=0.001时的TPR

TPR2：FPR=0.005时的TPR

TPR3：FPR=0.01时的TPR

最终成绩= 0.4 \* TPR1 + 0.3 \* TPR2 + 0.3 \* TPR3

**该评分指标的计算代码如下（需要用到pandas）:**

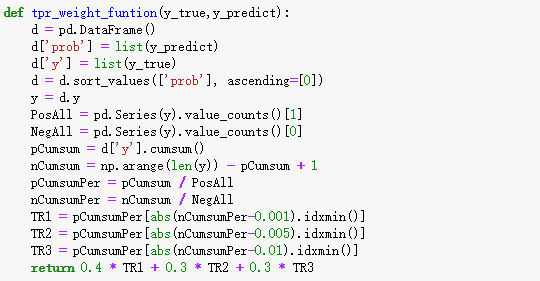


图 3-2 评分指标代码

# 四、认识数据

## 4.1特征字段的类型

在该课题中，结合业务背景，通常通过同类特征间的加减乘除可以构建一些新的特征。一般将数据的字段归为几类，这样便于理解字段，也为后面特征工程或是特征降维做准备。将操作详单和交易详单中的特征字段进行归类如下：

**（1）操作详单Operation**

时间日期：day, time

操作：mode, success

操作系统：os

版本：version

设备：device1, device2, device\_code1, device\_code2, device\_code3

ip：ip1, ip2, ip1\_sub, ip2\_sub

mac：mac1, mac2

wifi：wifi

地理位置：geo\_code

**（2）交易详单Transaction**

平台：channel

日期时间：day, time

资金：trans\_amt, bal, amt\_src1, amt\_src2

商户：merchant, code1, code2

交易类型：trans\_type1, trans\_typ2

账户：acc\_id1, acc\_id2

设备：device\_code1, device\_code2, device\_code3, device1, device2

ip：ip1, ip1\_sub

mac：mac1

地理位置：geocode

营销活动：market\_code, market\_type

## 4.2 结合业务的分析

分析业务有助于我们构建和选择特征。通过查询资料得知，电商和金融中羊毛党操作特点（后面的特征工程可以基于此）：

1. 同一个设备上有较多用户数。

2. 羊毛党都是机器操作，页面浏览、点击、用户交易时间、操作时间间隔要远小于普通较一般用户。

3. 单个用户参与活动频率，交易频次，提现频率、营销返现、刷订单等行为，都是羊毛党反欺诈评分模型的判断依据。

4. 用不同的IP登录不同的帐号。

5. 优惠券或营销活动有极低的用户转化率。

6. 同一个用户有较多的地理位置。

# 五、数据探索

进行可视化数据探索代码如下：

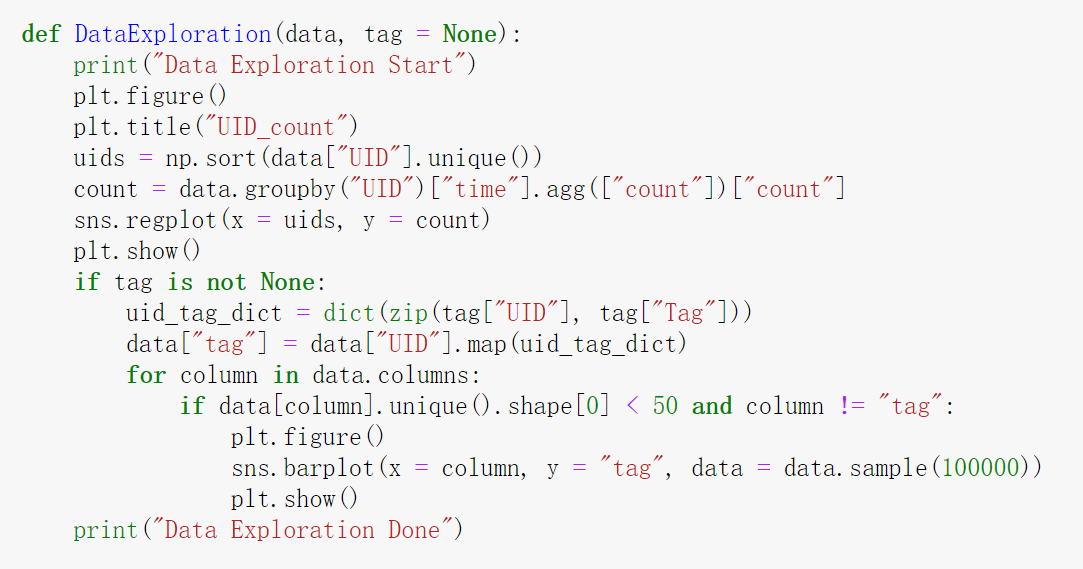


图 5-1 数据探索代码

首先用regplot画出x轴为UID，y轴为UID对应的记录数目的散点图。

然后遍历所有字段，如果该字段类别数小于50，画出该字段的条形图。

## 5.1可视化结果和结论

**（1）UID\_count（交易数据）**

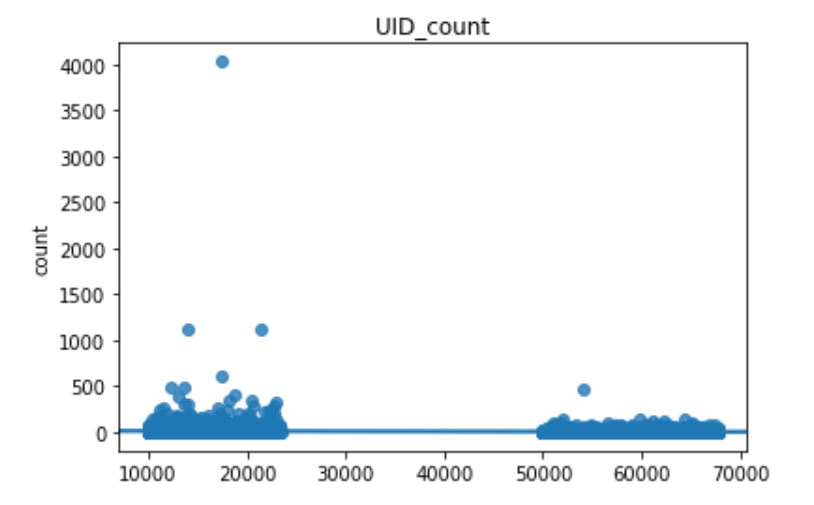


图 5-2 UID\_count

分析：UID为17520的用户交易次数远大于其他用户，是异常值。

**（2）day（操作数据/交易数据）**

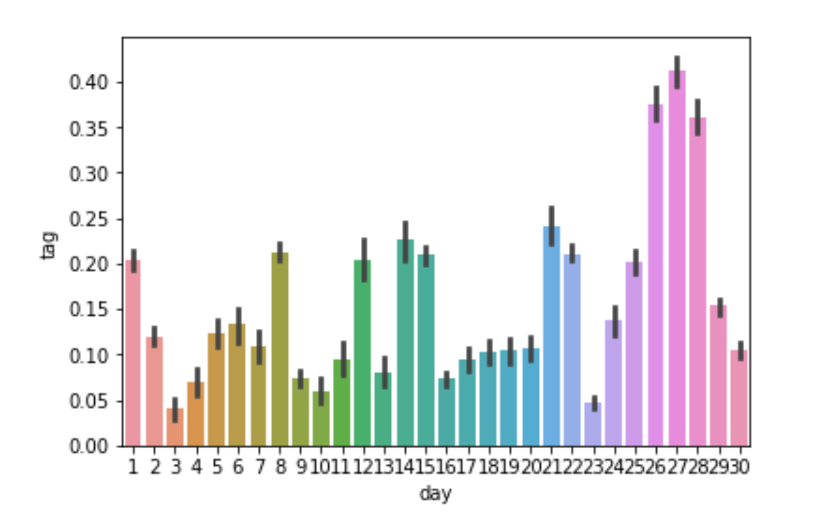
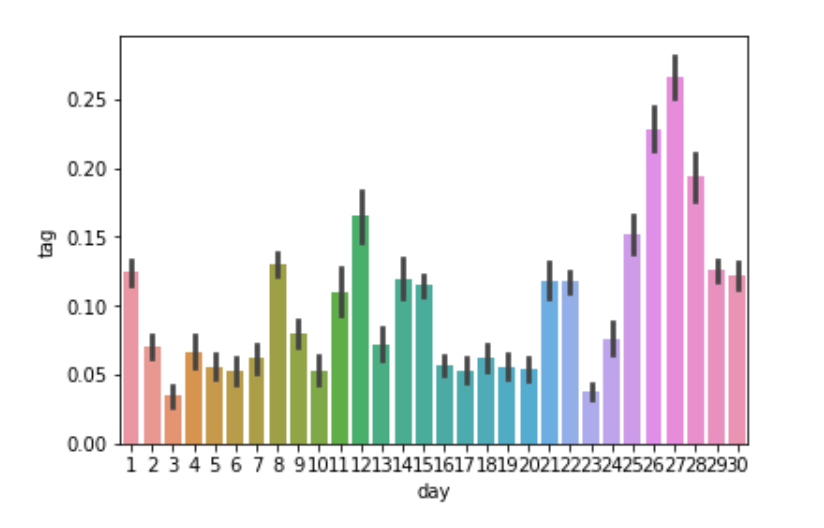


图 5-3 day\_tag\_operation 图 5-4 day\_tag\_transaction

分析：不同day的“羊毛党”比例不一致，可以创建特征day\_frequency代表用户的日操作频率/日交易频率。

**（3）success（操作数据）**

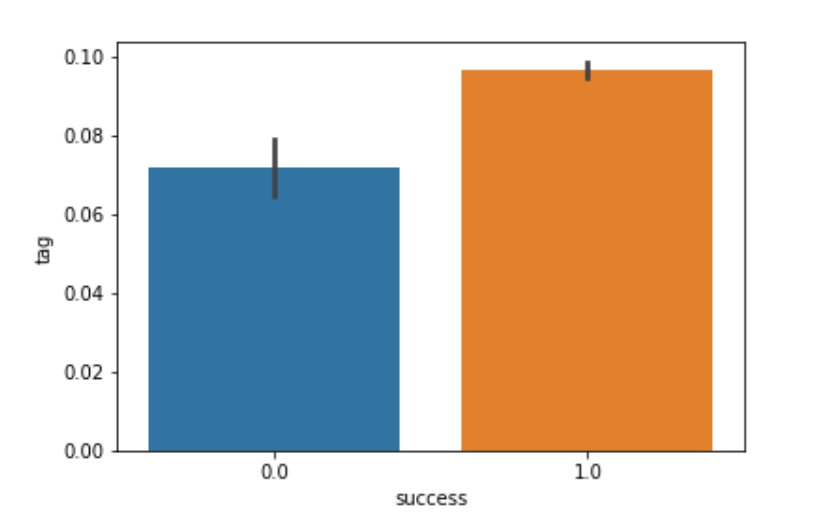


图 5-5 success\_tag

分析：success为1的“羊毛党”比例稍高，可以创建特征success\_mean特征，代表用户的平均操作成功率。

**（4）os（操作数据）**

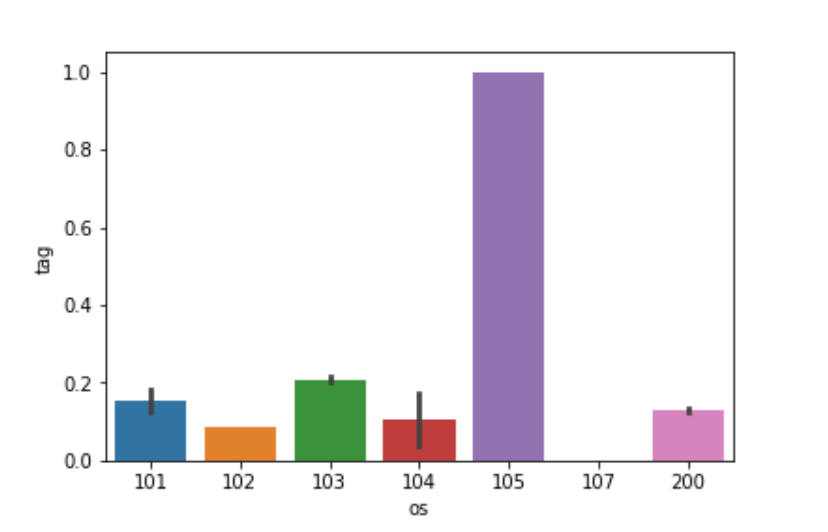


图 5-6 os\_tag

分析：os为105的用户“羊毛党“比例为100%，可以创建特征os\_has\_105，代表用户操作系统中是否有105；os为107的用户“羊毛党”极少，可以创建特征os\_has\_107，代表用户操作系统中是否有107。

**（5）channel（交易数据）**

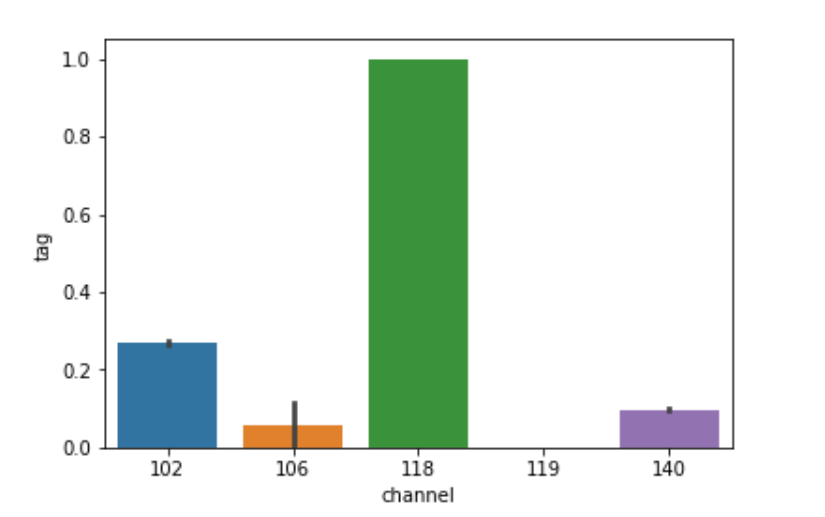


图 5-7 channel\_tag

分析：channel为118的用户“羊毛党”比例为100%，可以创建特征channel\_has\_118，代表用户平台中是否有118；channel为119的用户“羊毛党”极少，可以创建特征channel\_has\_119，代表用户平台中是否有119。

# 六、数据预处理

## 6.1数据预处理的一般方法

数据预处理一般会遇到以下问题：

1. 缺失值处理
2. 异常值处理
3. 重复值处理
4. 离散字段处理

在后文我们分别详细介绍。

## 6.2本次比赛数据的预处理

根据数据探索结果，本次竞赛主要做以下数据预处理：

1. 缺失值处理：由于后面要从原始数据中创建新特征，需要保留缺失值。
2. 异常值处理：

C:\Users\wanghui\Documents\WeChat Files\tuxiaodou-\Files\pics\代码图_outlier.PNG

1. 重复值处理：

C:\Users\wanghui\Documents\WeChat Files\tuxiaodou-\Files\pics\代码图_duplicates.PNG

1. 离散字段处理：通常是对离散字段进行one-hot编码，由于后面要根据离散字段创建新特征，需要保留离散字段。

# 七、特征工程

## 7.1特征创建（第一次）

1. 根据time字段创建hour字段，代表用户在哪一时段进行的操作或交易。

目的：为了创建用户的操作时段/交易时段以及在同一时段的操作频率/交易频率等特征，需要从time中提取出hour字段。

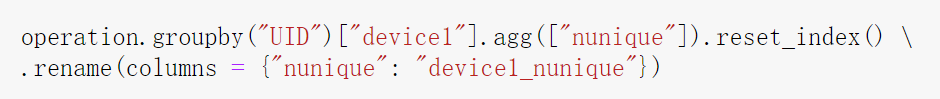
代码如下：

****

2. 对每个column，根据UID分组，创建特征column\_nunique，代表每个用户在该column上有多少种不同的取值。

例如：对于operation中的device1，创建特征device1\_nunique,代表每个用户使用过多少种不同的device1进行过操作。根据经验，“羊毛党”一般会有多个设备、ip等等，可以凭借这一特征对是否是“羊毛党”进行区分。

代码如下：



## 7.2特征选择

由于特征过多会引起所谓的“维度灾难”，增加计算量，降低模型泛化能力，所以我们要进行特征选择。

**特征选择的一般方法**

1. 过滤式选择，主要有方差选择法、相关系数法。
2. 包裹式选择，即递归特征消除法。
3. 嵌入式选择法。

**三种方法的比较：**

**过滤式：**根据特征的方差或相关系数对特征重要程度进行排序，方差越大、与目标变量相关系数越高，则特征越重要。

**包裹式：**两种方式，第一种是从少到多，每次从原始特征中选择一个使效果最好的特征添加到所选特征集中，直到添加任何特征不能明显提高模型效果；第二种是从多到少，每次从原始特征集删除一个特征使得模型效果下降最少，直到删除任何特征都会使得模型效果明显下降，所剩的特征就是所选特征集。

**嵌入式：**有些模型本身有对特征重要程度排序的机制，可以用这些模型进行特征选择。

由于嵌入式特征选择效果最好、代码量少，因此我们采用嵌入式特征选择**。**代码如下：

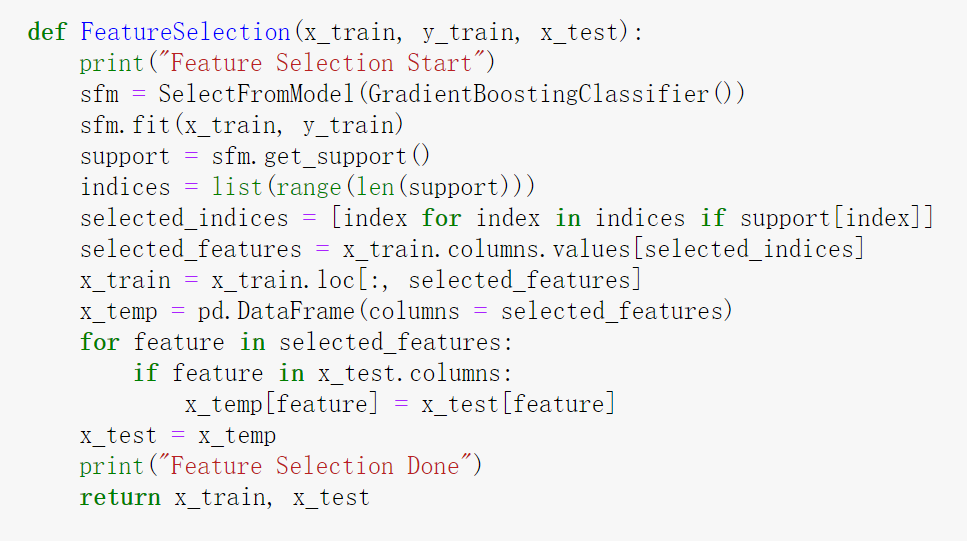
****

图 7-1 特征选择代码

创建SelectFromModel对象，用x\_train, y\_train拟合，用get\_support方法得到筛选后的特征。根据筛选后的特征，对训练集进行相应的处理。

# 八、模型优化

## 8.1调参方法

调参一般方法有GridSearch和RandomSearch。GridSearch是给定每个待调参数的几个选择，然后排列组合出所有可能性（就像网格一样），做Cross Validation，然后挑选出最好的那组参数组合。RandomSerach很类似，只是不直接给定参数的有限个取值可能，而是给出一个参数分布，从这个分布中随机采样一定个数的取值。

调参的时候需要理解这些参数到底是什么意思，如果过拟合了应该增大还是减小某个参数，这样才能有目的而不是盲目地调参。当然，想要找到最佳的参数很多时候需要一些经验和运气。也不需要极致追求最佳参数，大多数情况下找到一组相对不错的参数就可以了，往往还有别的方法来提升总成绩。

## 8.2课程模型与调参

本次课程使用模型：LR（逻辑回归），GBDT（梯度提升树），XGBoost

本次课程使用梯度提升决策树进行模型调参（GridSearchCV），即给定参数取值范围，通过梯度搜索的方式，得到最优参数。

## 8.3 XGBoost算法介绍

XGBoost(eXtreme Gradient Boosting)是Gradient Boosting算法的一个优化的版本。

### 8.3.1 XGBoost的优势

1. 正则化

2. 并行处理

3. 高度的灵活性

4. 允许用户自定义优化目标和评价标准

5. 内置处理缺失值的规则

6. 内置交叉验证

7. 基于上一轮的结果上继续训练

### 8.3.2 XGBoost的参数

XGBoost的作者把所有的参数分成了三类：

1. 通用参数：宏观函数控制。
2. Booster参数：控制每一步的booster(tree/regression)。
3. 学习目标参数：控制训练目标的表现。

下面类比GBM讲解。

1. **通用参数**

这些参数用来控制XGBoost的宏观功能。

1. booster[默认gbtree]

选择每次迭代的模型，有两种选择：

gbtree：基于树的模型

gbliner：线性模型

1. silent[默认0]

当这个参数值为1时，静默模式开启，不会输出任何信息。

一般这个参数就保持默认的0，因为这样能帮我们更好地理解模型。

1. nthread[默认值为最大可能的线程数]

这个参数用来进行多线程控制，应当输入系统的核数。

如果你希望使用CPU全部的核，那就不要输入这个参数，算法会自动检测它。

还有两个参数，XGBoost会自动设置，目前你不用管它。

1. **booster参数**

尽管有两种booster可供选择，这里只介绍tree booster，因为它的表现远远胜过linear booster，所以linear booster很少用到。

1. eta[默认0.3]

和GBM中的 learning rate 参数类似。

通过减少每一步的权重，可以提高模型的鲁棒性。

典型值为0.01-0.2。

1. min\_child\_weight[默认1]

决定最小叶子节点样本权重和。

它和GBM的 min\_child\_leaf 参数类似，但不完全一样。

XGBoost的这个参数是最小样本权重的和，而GBM参数是最小样本总数。

这个参数用于避免过拟合。当它的值较大时，可以避免模型学习到局部的特殊样本。

但是如果这个值过高，会导致欠拟合。这个参数需要使用CV来调整。

1. max\_depth[默认6]

和GBM中的参数相同，这个值为树的最大深度。

这个值也是用来避免过拟合的。max\_depth越大，模型会学到更具体更局部的样本。

需要使用CV函数来进行调优。

典型值：3-10。

1. max\_leaf\_nodes

树上最大的节点或叶子的数量。

可以替代max\_depth的作用。

1. gamma[默认0]

在节点分裂时，只有分裂后损失函数的值下降了，才会分裂这个节点。

gamma指定了节点分裂所需的最小损失函数下降值。

这个参数的值越大，算法越保守。这个参数的值和损失函数息息相关，所以是需要调整的。

1. max\_delta\_step[默认0]

这参数限制每棵树权重改变的最大步长。

如果这个参数的值为0，那就意味着没有约束。

如果它被赋予了某个正值，那么它会让这个算法更加保守。

通常，这个参数不需要设置。但是当各类别的样本十分不平衡时，它对逻辑回归是很有帮助的。

1. subsample[默认1]

和GBM中的subsample参数一模一样。

这个参数控制对于每棵树，随机采样的比例。

减小这个参数的值，算法会更加保守，避免过拟合。但是，如果这个值设置得过小，它可能会导致欠拟合。

典型值：0.5-1。

1. colsample\_bytree[默认1]

和GBM里面的max\_features参数类似。

用来控制每棵随机采样的列数的占比(每一列是一个特征)。

典型值：0.5-1

1. colsample\_bylevel[默认1]

用来控制树的每一级的每一次分裂，对列数的采样的占比。

subsample参数和colsample\_bytree参数可以起与该参数相同的作用。

1. clambda[默认1]

权重的L2正则化项 (和Ridge regression类似)。

这个参数是用来控制XGBoost的正则化部分的。

这个参数在减少过拟合上可以挖掘出更多用处。

1. alpha[默认1]

权重的L1正则化项 (和Lasso regression类似)。

可以应用在很高维度的情况下，使得算法的速度更快。

1. scale\_pos\_weight[默认1]

在各类别样本十分不平衡时，把这个参数设定为一个正值，可以使算法更快收敛。

**(3)学习目标参数**

这个参数用来控制理想的优化目标和每一步结果的度量方法。

1. objective[默认reg:linear]

这个参数定义需要被最小化的损失函数。最常用的值有：

binary: logistic 二分类的逻辑回归，返回预测的概率(不是类别)。

multi: softmax 使用softmax的多分类器，返回预测的类别(不是概率)。

在这种情况下，你还需要多设一个参数：num\_class(类别数目)。

multi:softprob 和multi:softmax参数一样，但是返回的是每个数据属于各个类别的概率。

1. eval\_metric[默认值取决于objective参数的取值]

对于有效数据的度量方法。

对于回归问题，默认值是rmse，对于分类问题，默认值是error。

典型值有：

rmse 均方根误差

mae 平均绝对误差

logloss 负对数似然函数值

error 二分类错误率(阈值为0.5)

merror 多分类错误率

mlogloss 多分类logloss损失函数

auc 曲线下面积

1. seed(默认0)

随机数的种子。

设置它可以复现随机数据的结果，也可以用于调整参数。

Python的XGBoost模块有一个sklearn包，XGBClassifier。这个包中的参数是按sklearn风格命名的。会改变的函数名是：

1、eta -> learning\_rate

2、lambda -> reg\_lambda

3、alpha -> reg\_alpha

# 九、模型评估

## 9.1模型评估记录（第一次）

记录特征、模型以及对应的roc\_auc\_score, tpr\_weight\_score。

模型评估的记录如下：

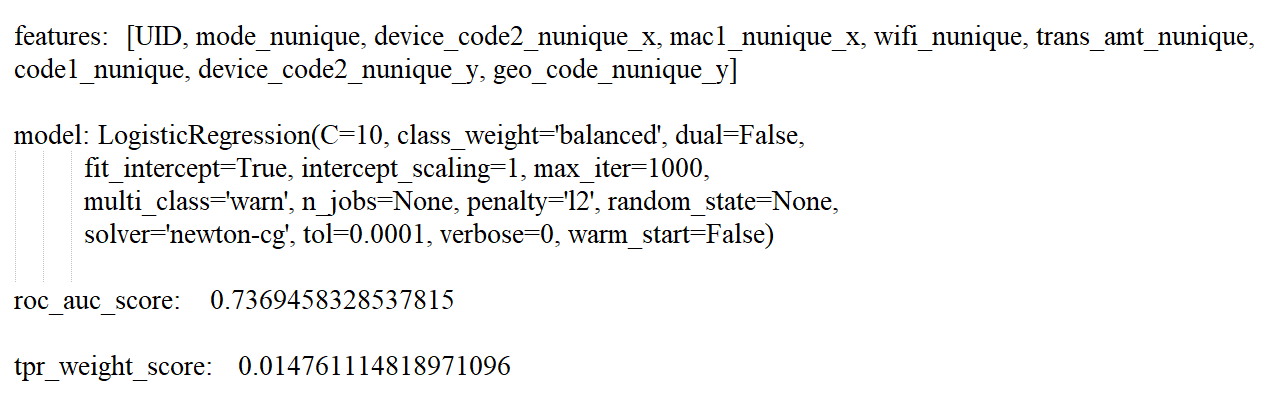


图 9-1 LR\_record1

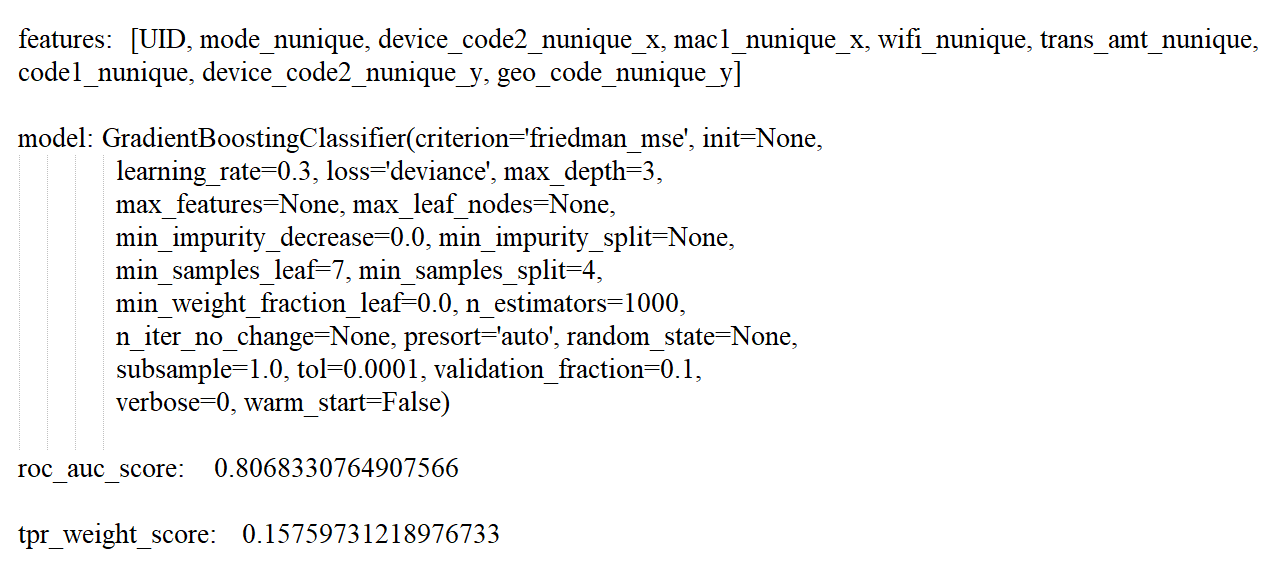


图 9-2 GBDT\_record1

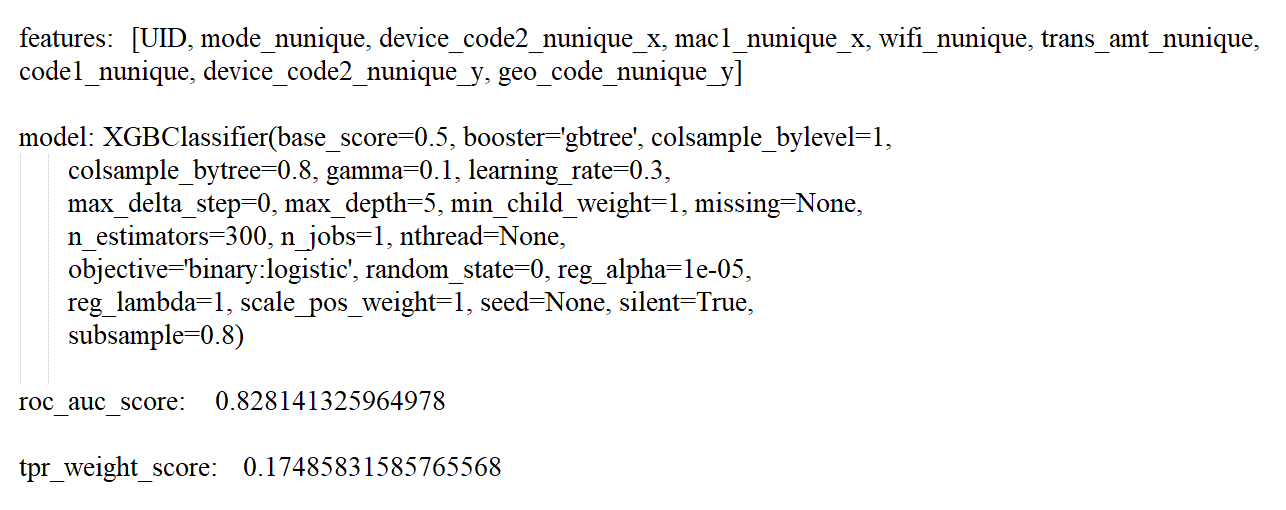


图 9-3 XGBoost\_record1

# 十、重复七、八、九步直到模型效果无明显提升

## 10.1特征工程（第二次）

对每个column创建特征：

column\_nunique\_UID\_max,

column\_nunique\_UID\_min,

column\_nunique\_UID\_mean,

column\_count\_UID\_max,

column\_count\_UID\_min,

column\_count\_UID\_max

代码如下：

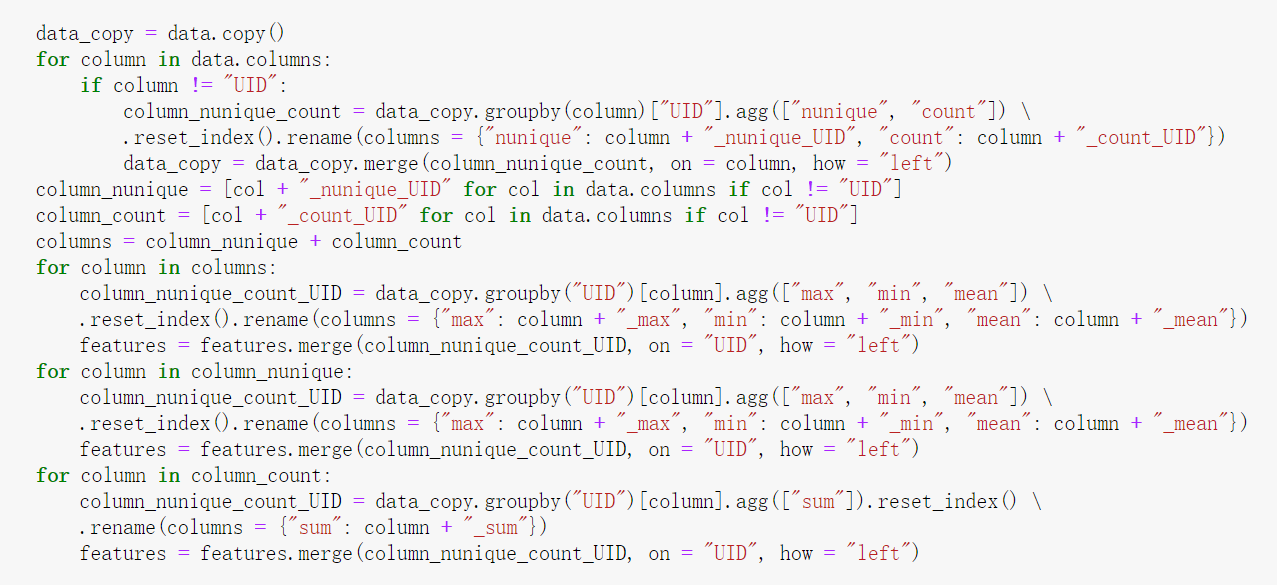


图 10-1 特征工程代码（第二次）

以Device1为例:

1. 先创建字段device1\_nunique\_UID:

代表每个设备出现在多少个不同的用户的操作/交易中。根据device1\_nunique\_UID创建特征：

device1\_nunique\_UID\_max,

device1\_nunique\_UID\_min,

device1\_nunique\_UID\_mean,

代表每个用户所用设备最多、平均、最少被多少个不同的用户使用过。

1. 创建字段device1\_count\_UID:

代表每个设备出现过多少次操作/交易中。

根据device1\_count\_UID创建特征

device1\_count\_UID\_max,

device1\_count\_UID\_min,

device1\_count\_UID\_mean,

代表每个用户所用设备被所有用户使用的次数的最大值、最小值、平均值。

根据经验，用户的设备被越多不同用户使用，使用的次数越多，那么该用户就越有可能是“羊毛党”。

其他字段类似。

## 10.2模型评估的记录（第二次）

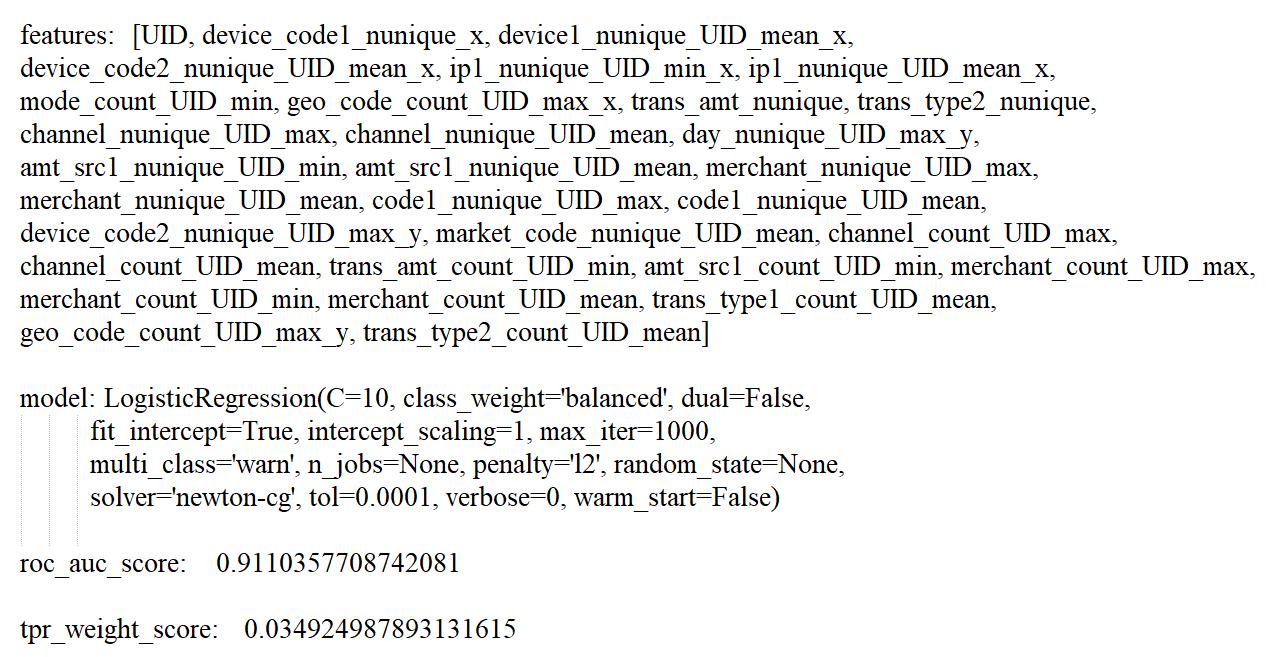


图 10-2 LR\_record2

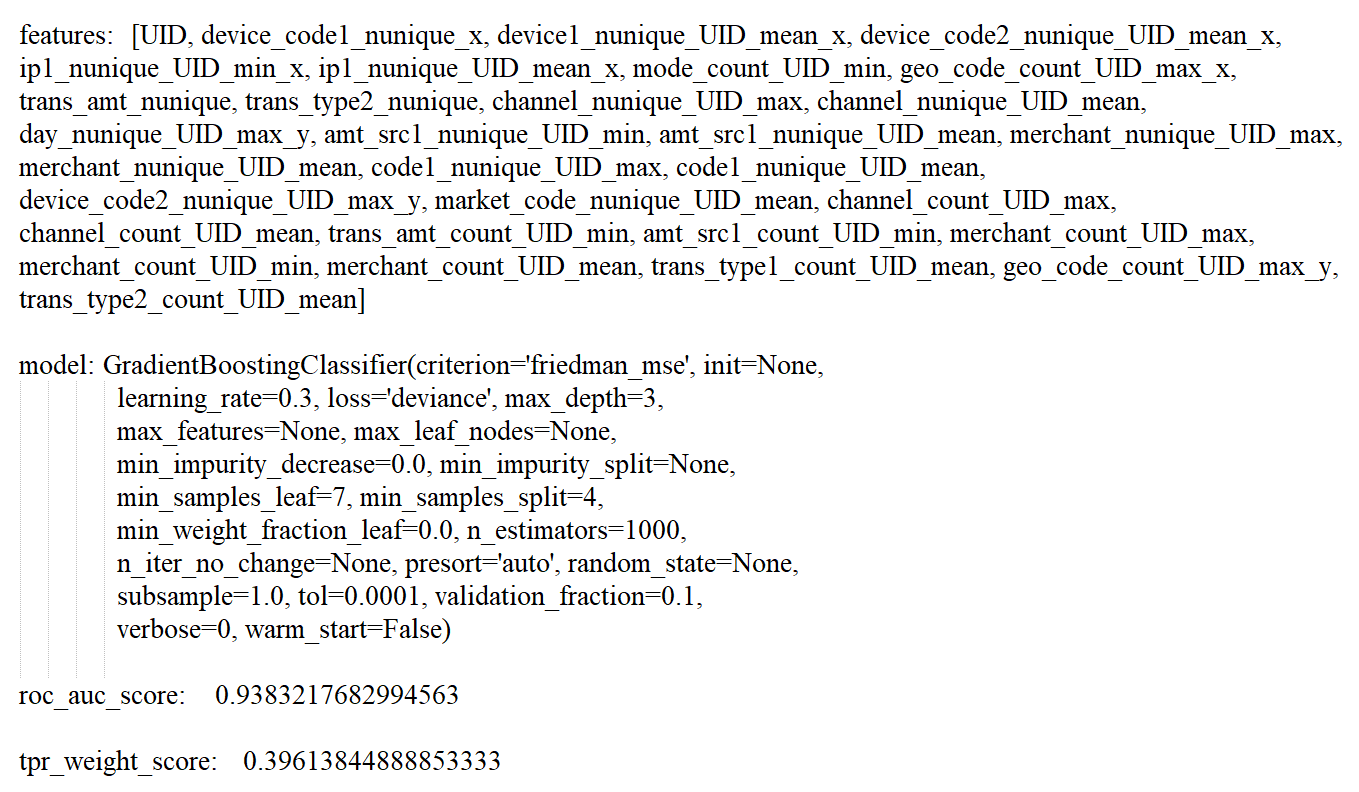


图 10-3 GBDT\_record2

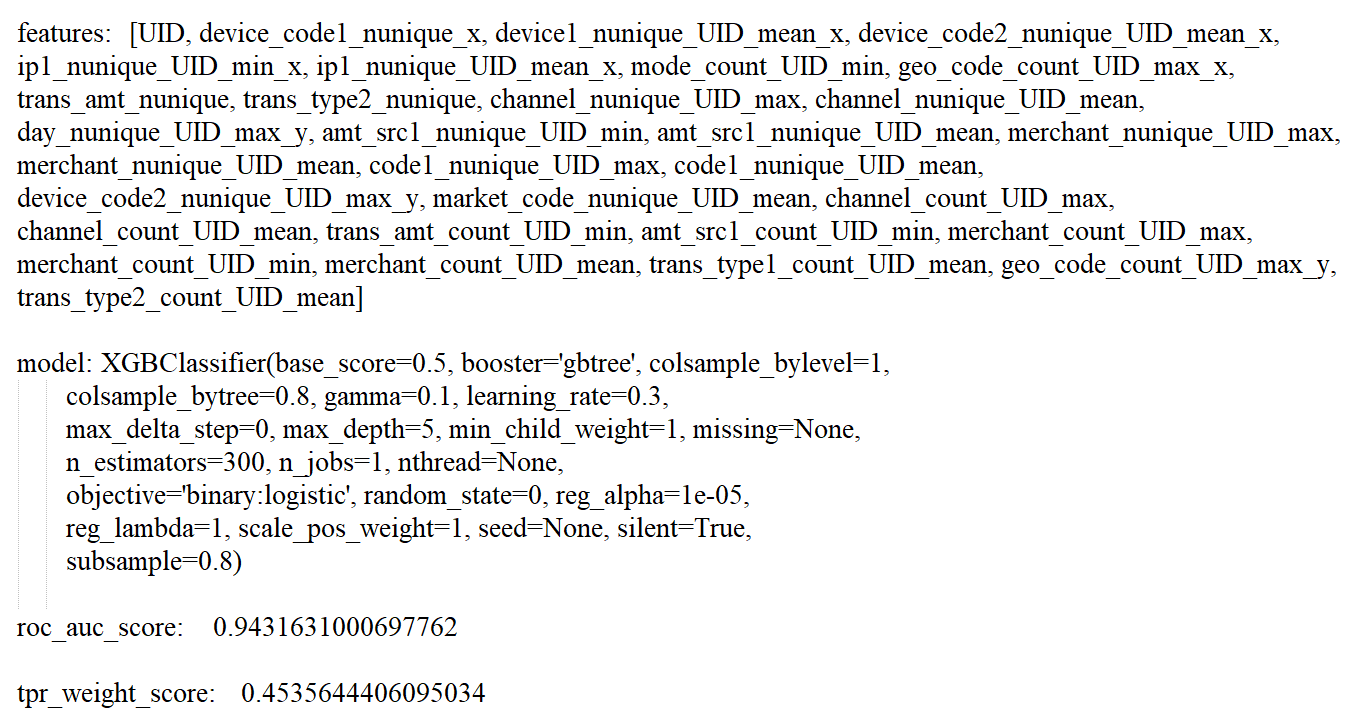


图 10-4 XGBoost\_record2

## 10.3特征工程（第三次）

根据数据探索的结论，创建特征day\_frequency, hour\_frequency，代码如下：

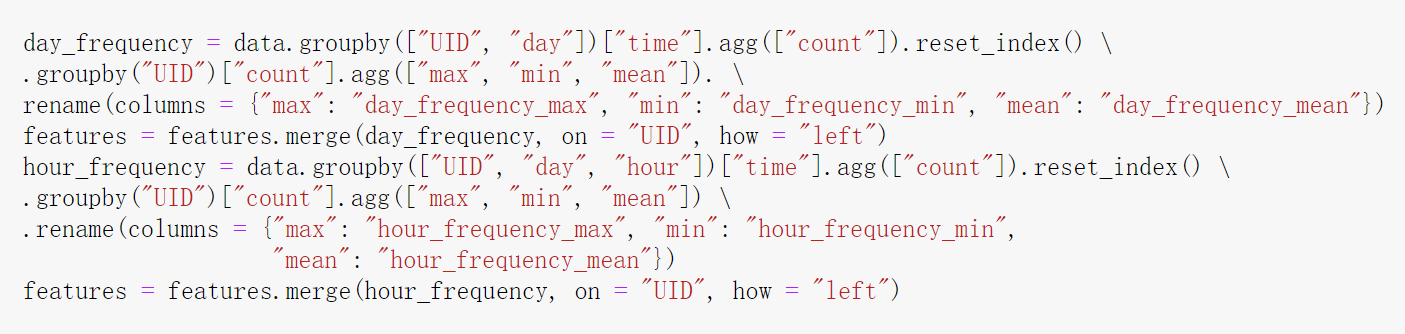


图 10-5创建 day\_ hour\_frequency

特征success\_mean：

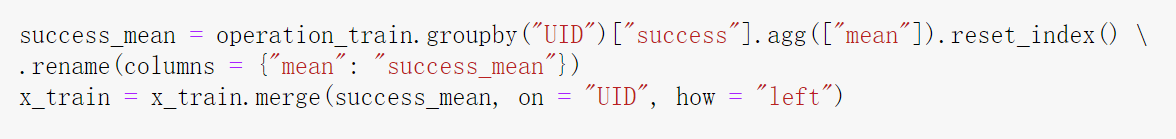


图 10-6 创建 success\_mean

特征os\_has\_105, os\_has\_107, channel\_has\_118, channel\_has\_119，特征os\_has\_105为例：

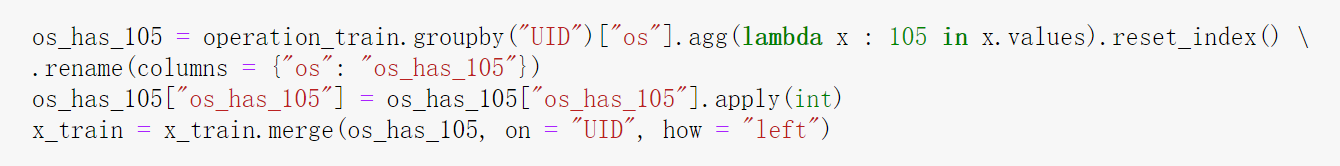


图 10-7 创建os\_has\_105

os\_has\_107, channel\_has\_118, channel\_has\_119代码类似。

## 10.4模型评估的记录（第三次）

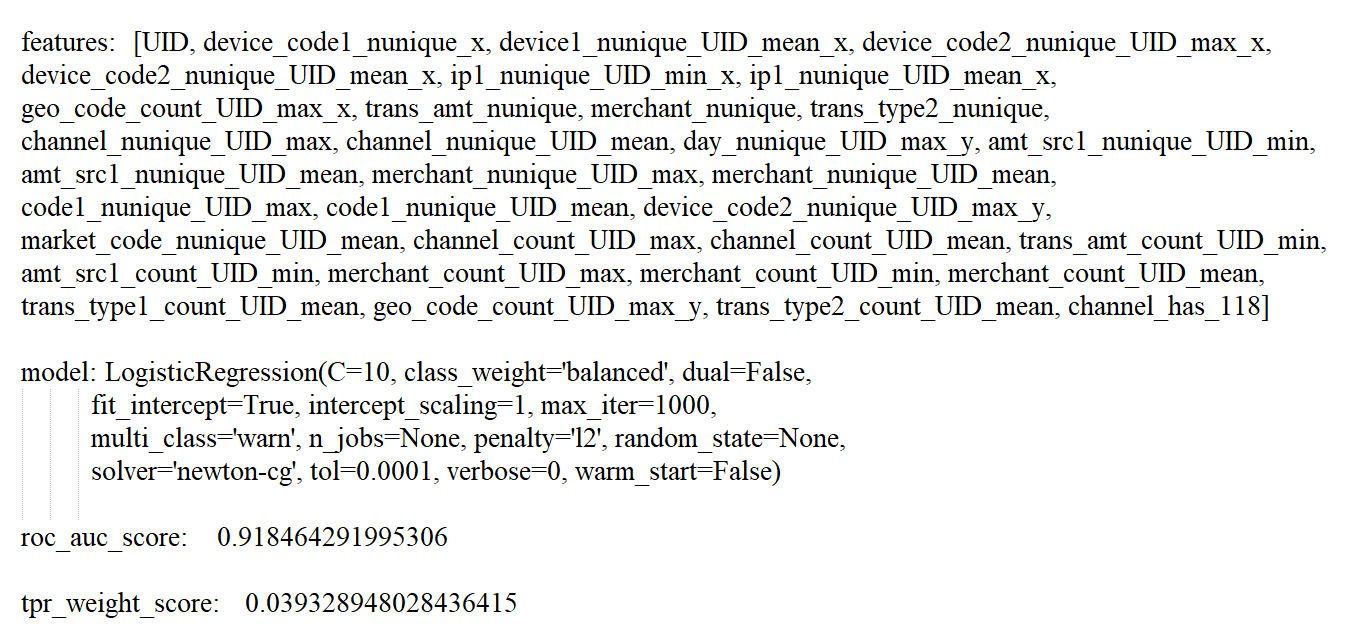


图 10-8 LR\_record3

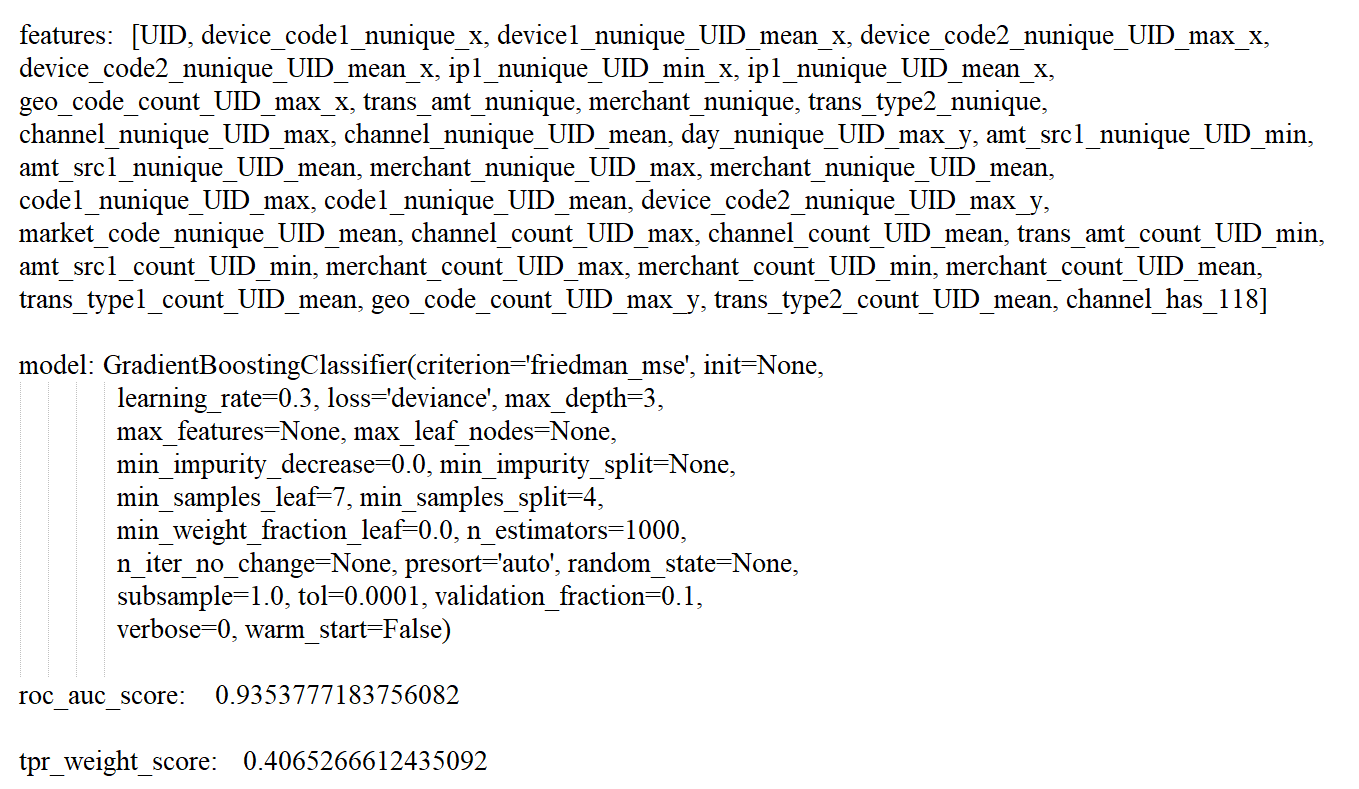
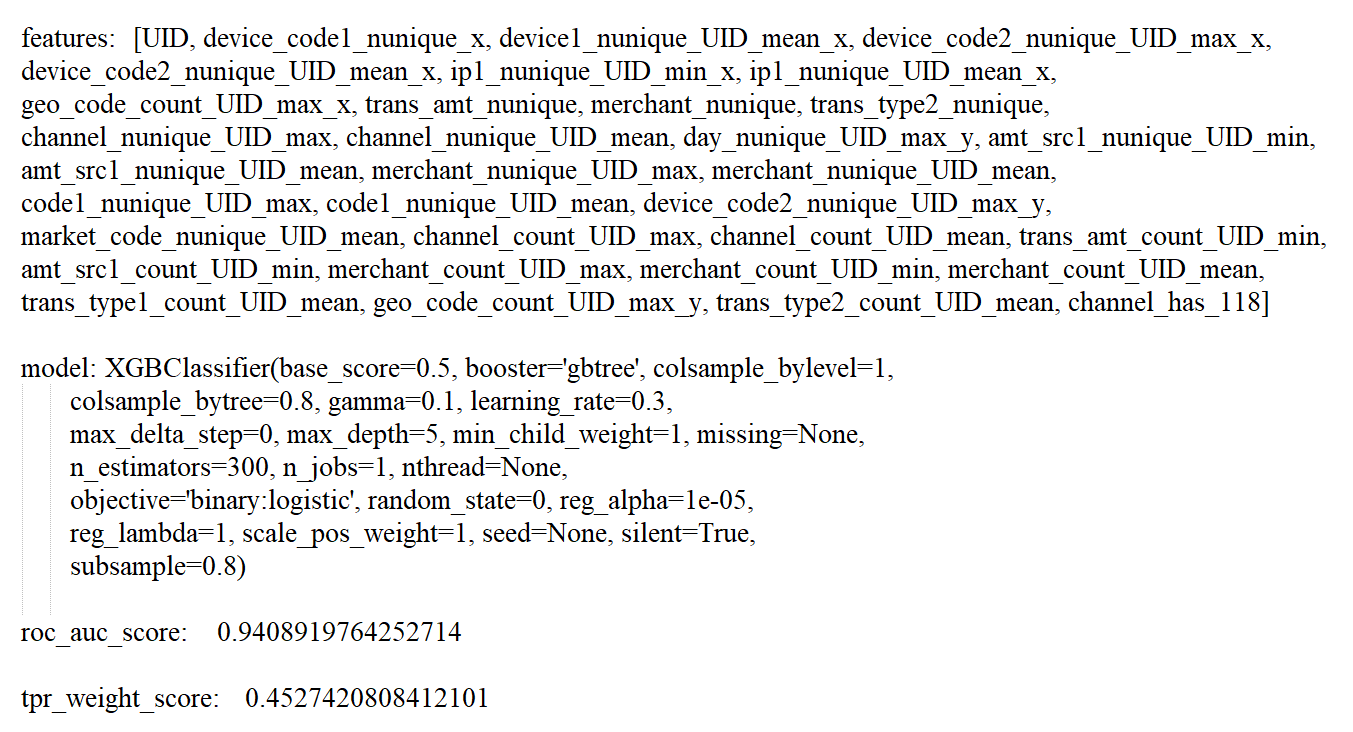


图 10-9 GBDT\_record3



**图 10-10 XGBoost\_record3**

# 十一、模型融合

模型融合在比赛中非常重要，同时也是一个很大的话题。两种比较有效的方式Bagging和Stacking。

## 11.1 Bagging

Bagging使用训练数据的不同随机子集来训练每个 Base Model，最后进行每个 Base Model 权重相同的 Vote。也即 Random Forest 的原理。每个模型的权重怎么算？一般是根据单个模型的表现好坏来决定，可以看测试集上的表现，Kaggle比赛里可以看LB Score，但是也不能完全看Public LB，这样就过拟合了。

Averaging是其中一种，是最简单粗暴也是最好理解的模型融合方式，而且效果还挺好的。实际上就是加权平均。虽然简单，但是非常有效。如果模型的多样性足够，比如有的模型擅长从税收角度预测房价，有的模型擅长从房间数量来预测房价，把这些模型平均后，取长补短，就能获得一个更准确泛化能力更强的模型。

## 11.2 Stacking

Stacking应该是目前各类竞赛中最好用的模型融合方法了。看下面这张流传很广的图，其实Stacking并不难理解。

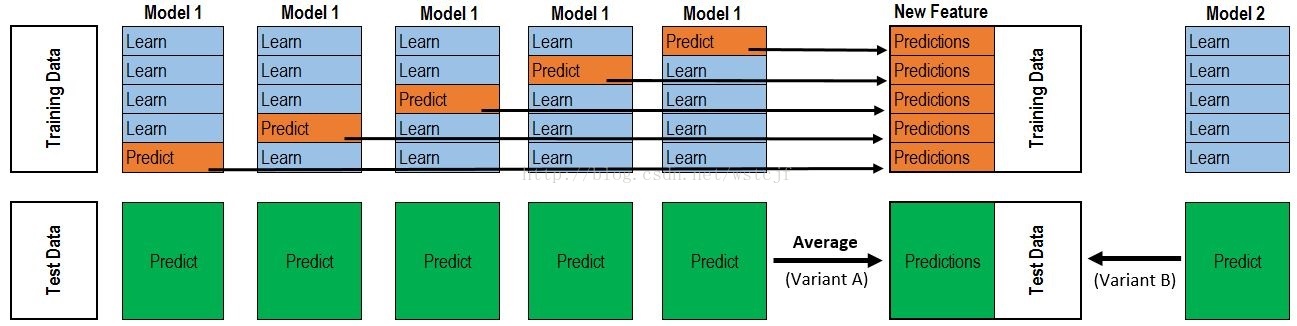
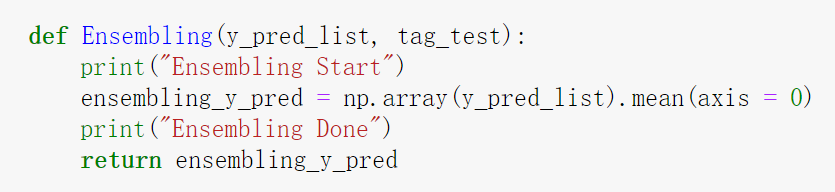


图11-1 Stacking图解

Stacking的核心思想是把第一层模型的结果作为第二层模型的特征，然后训练第二层模型得到最终结果。以5-fold stacking为例，将训练集随机分成5份，分别用其中4份作训练，来预测剩下的1份，同时也预测所有的测试集。这样，一个模型训练了五次，对训练集的预测拼起来，正好每一个训练集的样本都有一个预测值。对测试集的每个样本，则有5个预测值，求平均值作为测试集的预测值。这样，训练集和测试集都有一个预测值，作为第二层模型的特征。

**融合方法：我们使用融合方法是Bagging中的简单平均。**

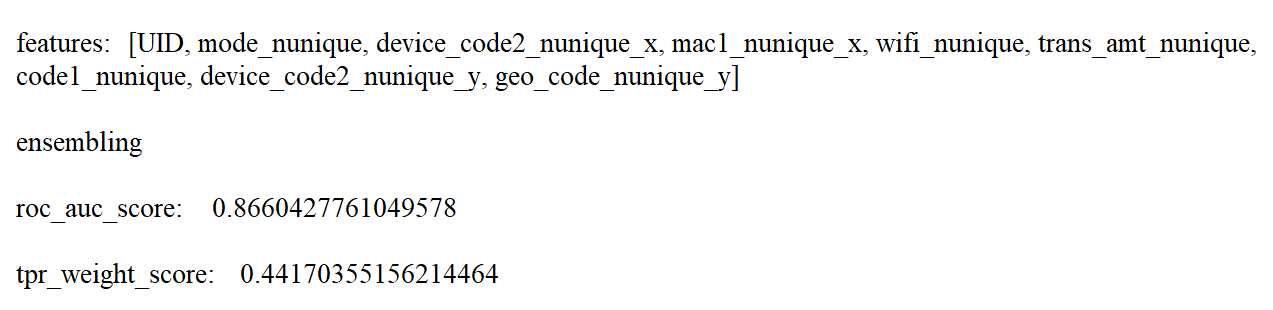
代码如下：

****

**图 11-2 模型融合**

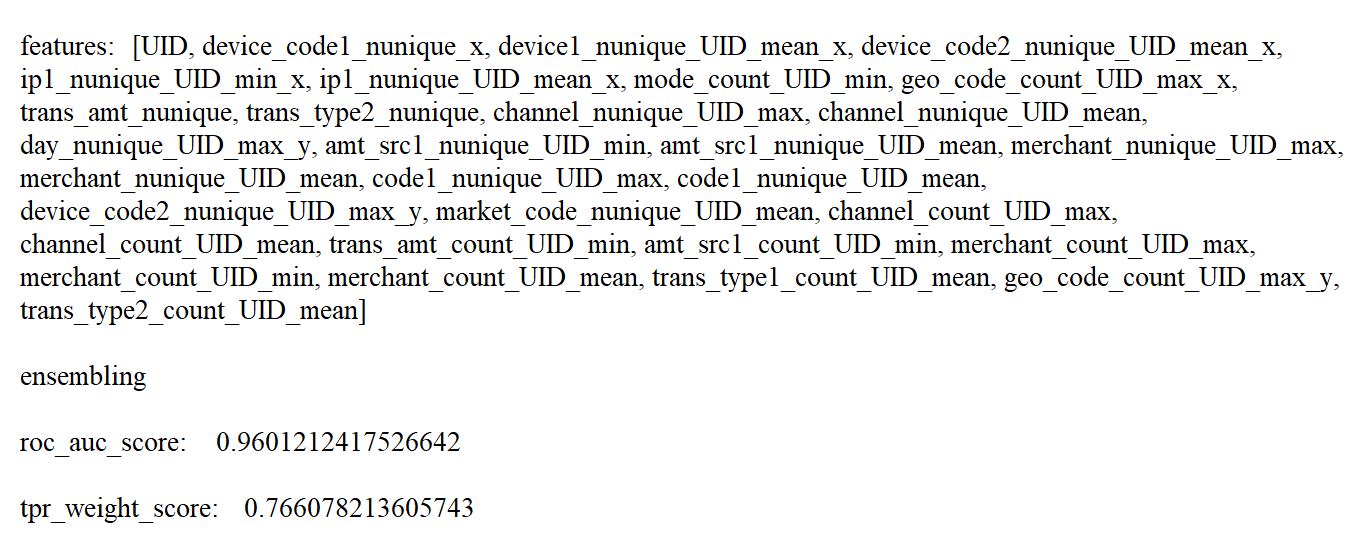
**融合效果**

第一次特征工程：

****

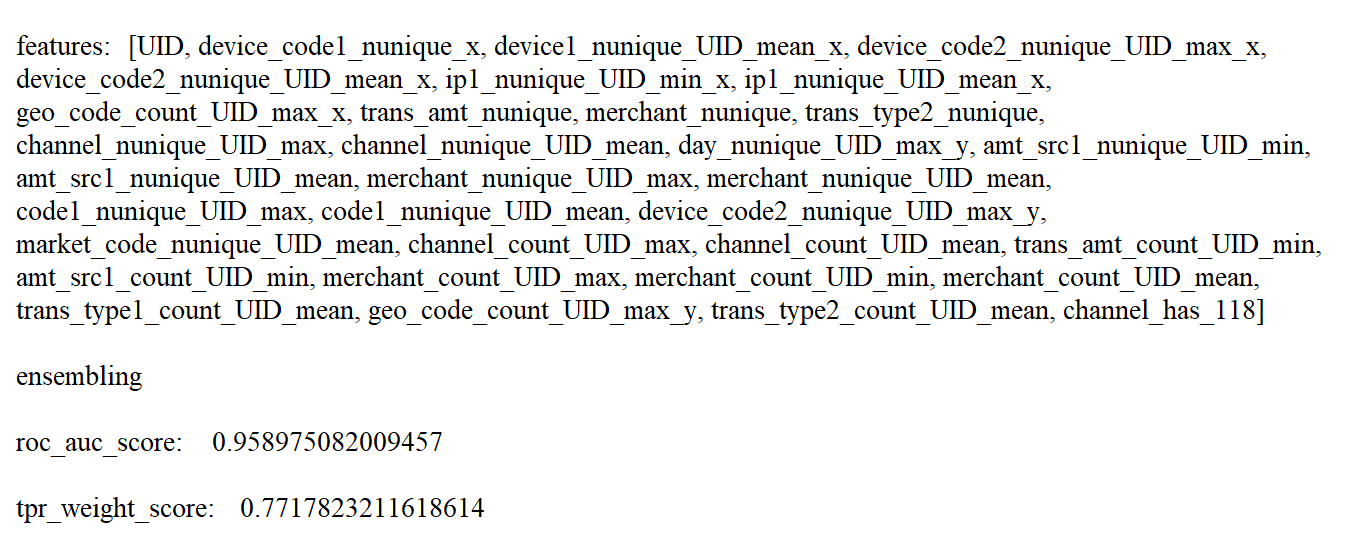
**图 11-3 Ensembling\_record1**

第二次特征工程：

****

**图 11-4 Ensembling\_record2**

第三次特征工程：

****

**图 11-5 Ensembling\_record3**

# 十二、提交

建立代码的同级文件夹三个：Data、Submission、Records

Data：用于存放竞赛需要的数据

Submission：用于存放提交的数据

Records：用于存放跑数据中间的记录