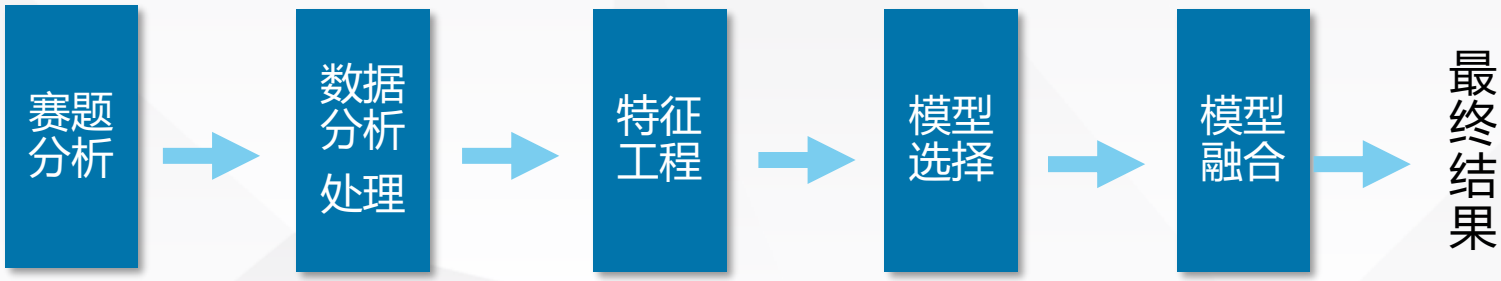
数据挖掘比赛案例介绍

甜橙金融杯大数据竞赛

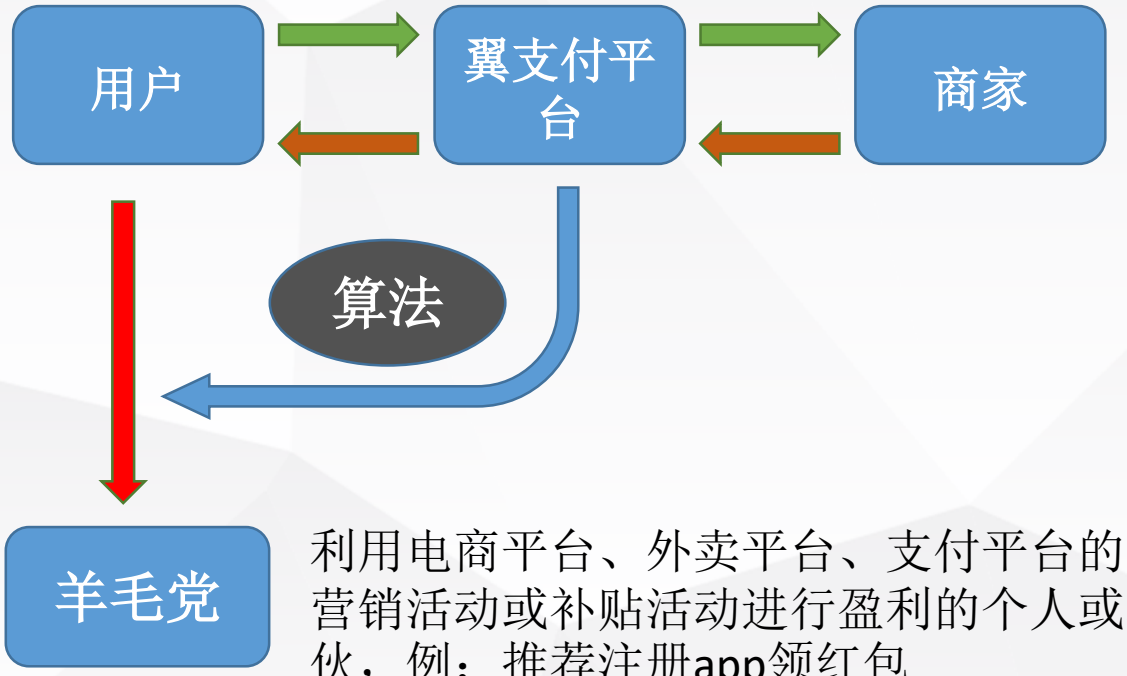
基本流程



CONTANTS

甜橙杯赛题背景

支付平台应用机器学习算法识别羊毛党用户。



利用电商平台、外卖平台、支付平台的一些营销活动或补贴活动进行盈利的个人或者团伙，例：推荐注册app领红包

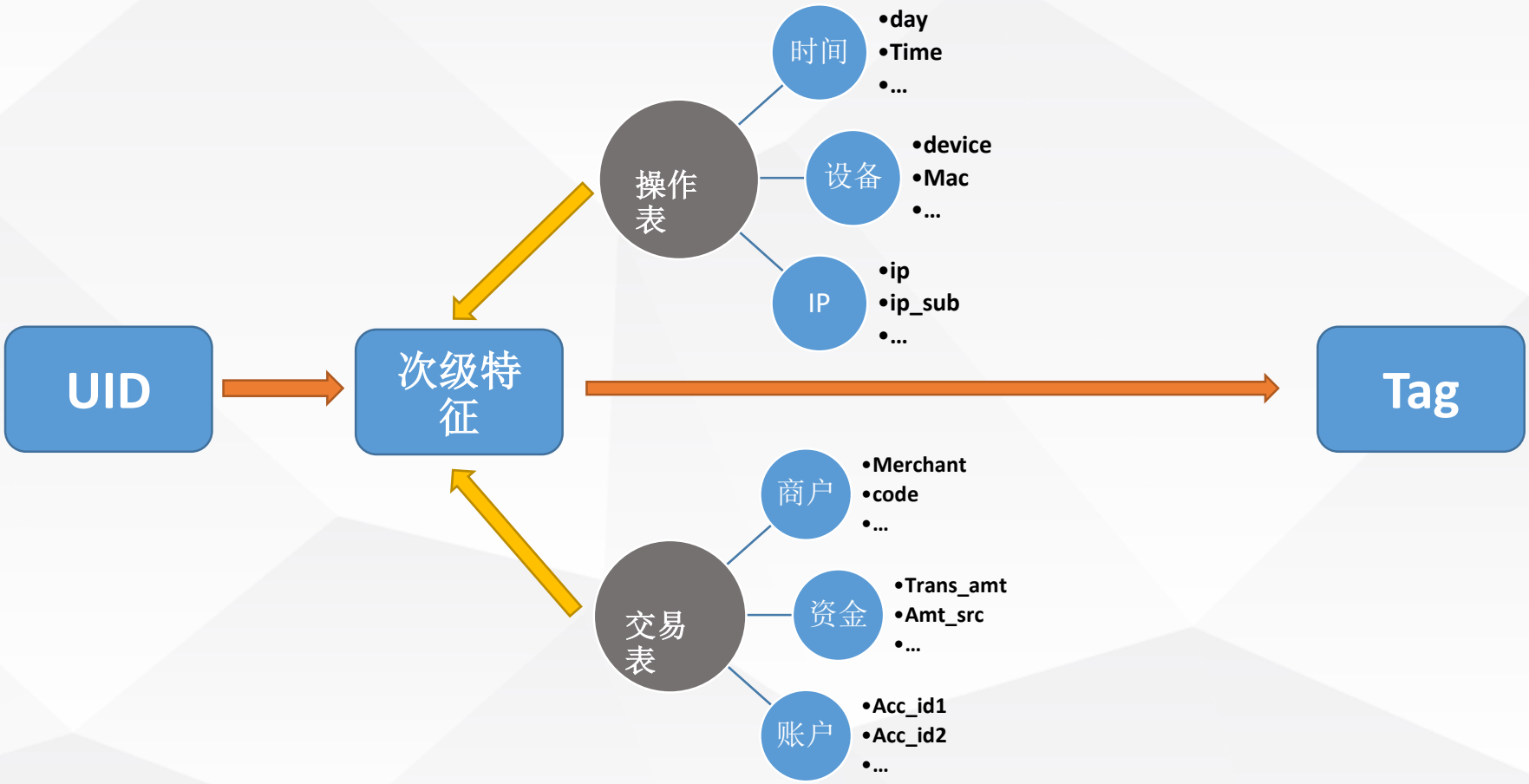
操作详单数据字典

字段名	中文解释	字段说明
UID	用户编号	
day	操作日期	连续的日期标识， E.g. 1为第一天，2为第二天，以此类推
mode	操作类型	操作类型（例如：修改密码、查询余额…）
success	操作状态	
time	操作时间点	
os	操作系统	
version	客户端版本号	
device1	操作设备参数1	设备名称加密，原字段如“Jack’s iphone”
device2	操作设备参数2	设备型号
device_code1	操作设备唯一标识1	设备号唯一标识加密，可用于安卓类设备的唯一标识
device_code2	操作设备唯一标识2	设备号唯一标识加密，可用于安卓类设备的唯一标识
mac1	MAC地址	操作设备MAC地址编码加密， 原字段如“38:XX:XX:XX:XX:92”
ip1	IP地址	操作设备IP地址编码加密
ip2	IP地址	操作电脑IP地址编码加密
device_code3	操作设备唯一标识3	设备号唯一标识加密，可用于苹果类设备的唯一标识
mac2	MAC地址	WIFI MAC地址编码加密， 原字段如“02:XX:XX:XX:XX:03
wifi	WIFI名称	WIFI名称，原字段如“A的wifi”
geo_code	地理位置	经纬度GeoHash编码
ip1_sub	IP地址	前三位操作设备IP地址编码加密（ip1前三位IP地址） 比如，原字段为12，34，56，7和12，34，56，8的ip地址前三位都为12，34，56，故脱敏后的值是一样的
ip2_sub	IP地址	前三位操作电脑IP地址编码加密（ip2前三位IP地址）

交易详单数据字典

字段名	中文解释	字段说明
UID	用户编号	
channel	平台	平台类型
day	交易日期	连续的日期标识， 1为第一天，2为第二天，以此类推
time	交易时间点	
trans_amt	脱敏后交易金额	保留大小关系
amt_src1	资金类型	交易资金源类型，例如“余额”、“银行卡”
merchant	商户标识	商户编码加密
code1	商户标识	商户子门店编码加密
code2	商户终端设备标识	商户交易终端设备编码加密
trans_type1	交易类型1	交易类型，例如“消费”，“退款”
acc_id1	账户相关	用户交易账户号编码加密
device_code1	操作设备唯一标识1	设备号唯一标识加密，可用于安卓类设备的唯一标识
device_code2	操作设备唯一标识2	设备号唯一标识加密，可用于安卓类设备的唯一标识 （唯一标识码并不会只是一种 但都能达到效果）
device_code3	操作设备唯一标识3	设备号唯一标识加密，可用于苹果类设备的唯一标识
device1	操作设备参数1	设备名称加密，原字段如“Jack’s iphone”
device2	操作设备参数2	设备型号
mac1	MAC地址	操作设备MAC地址编码加密， 原字段如“38:XX:XX:XX:XX:92”
ip1	IP地址	操作设备IP地址编码加密
bal	脱敏后账户余额	保留大小关系
amt_src2	资金类型	交易资金源类型，与1类型相似，2对银行卡做了细分
acc_id2	账户相关	转账操作的转出账户号编码加密
acc_id3	账户相关	转账操作的转入账户号编码加密
geocode	地理位置	经纬度GeoHash编码
trans_type2	交易类型2	交易类型，例如“线上”、“线下” trans_type2与trans_type1的维度和侧重不同
market_code	营销活动号编码	营销活动号编码加密
market_type	营销活动标识	营销活动类型
ip1_sub	IP地址	前三位操作设备IP地址编码加密（ip1前三位IP地址）

赛题任务



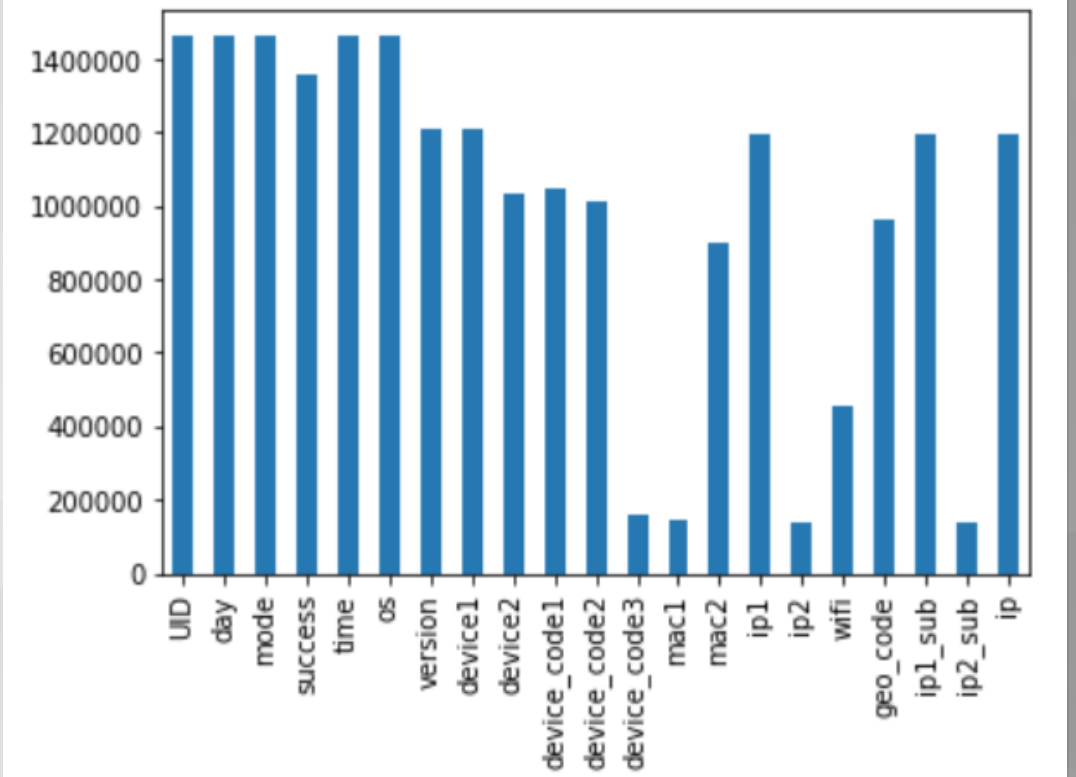
通过训练学习用户在消费过程中的关联操作、交易详单信息，来识别“羊毛党”

二分类问题

数据探索-缺失值

```
1 op_data.count().plot(kind = 'bar')
```

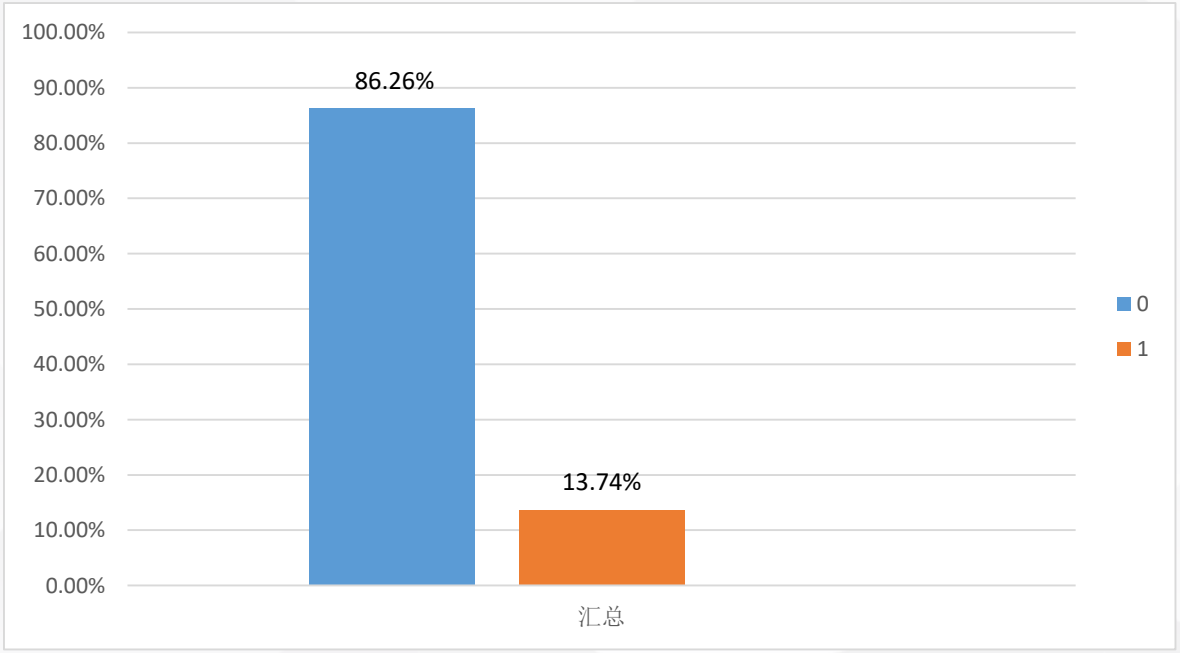
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f043860c990>
```



1. 删除字段
2. 数据补齐
3. 直接利用

数据探索

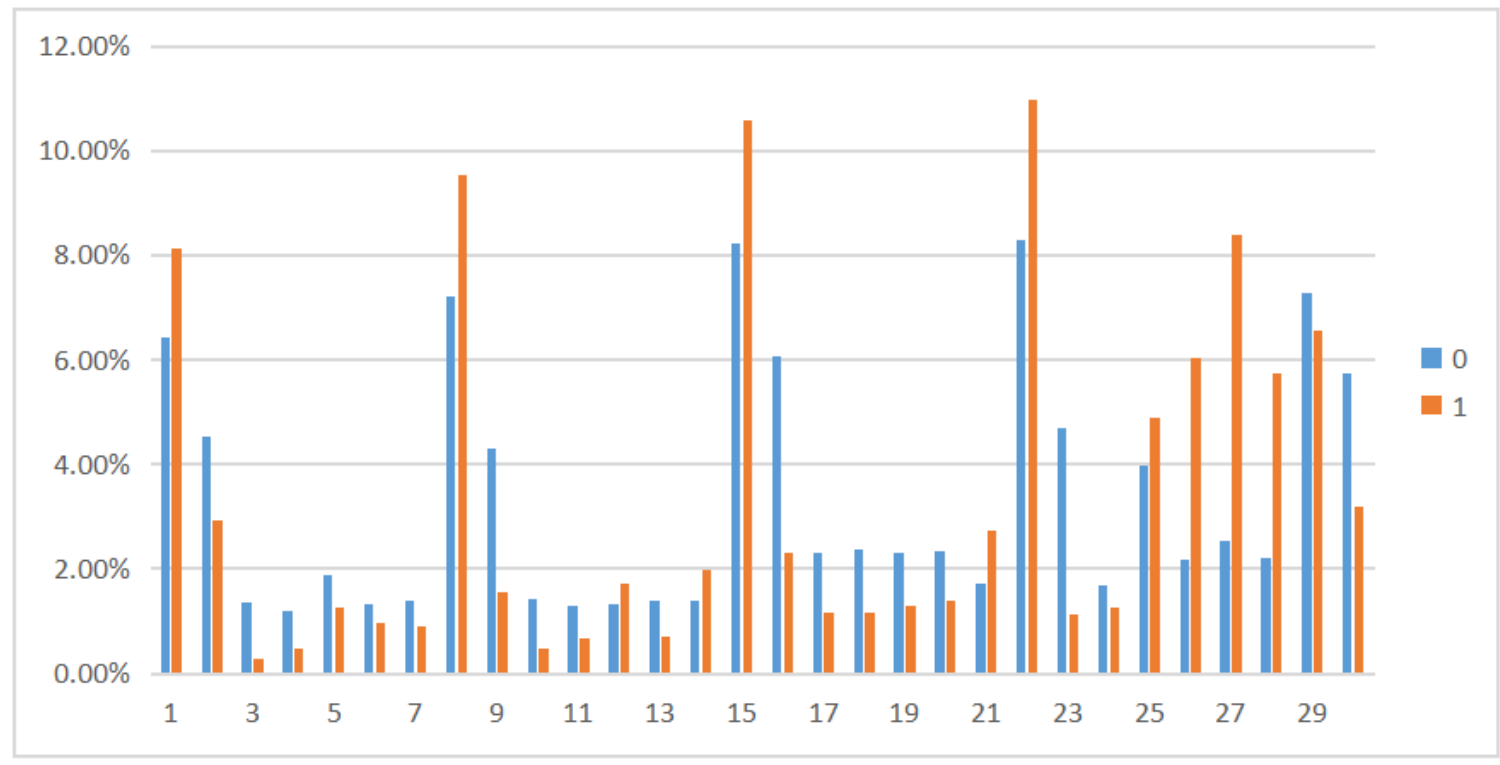
训练集正负样本比例



训练集里面的正样本的比例只有不到14%，这样的样本是非常不平衡的，我们后面的模型参数的设置和模型融合部分都需要参考这个比例

数据探索

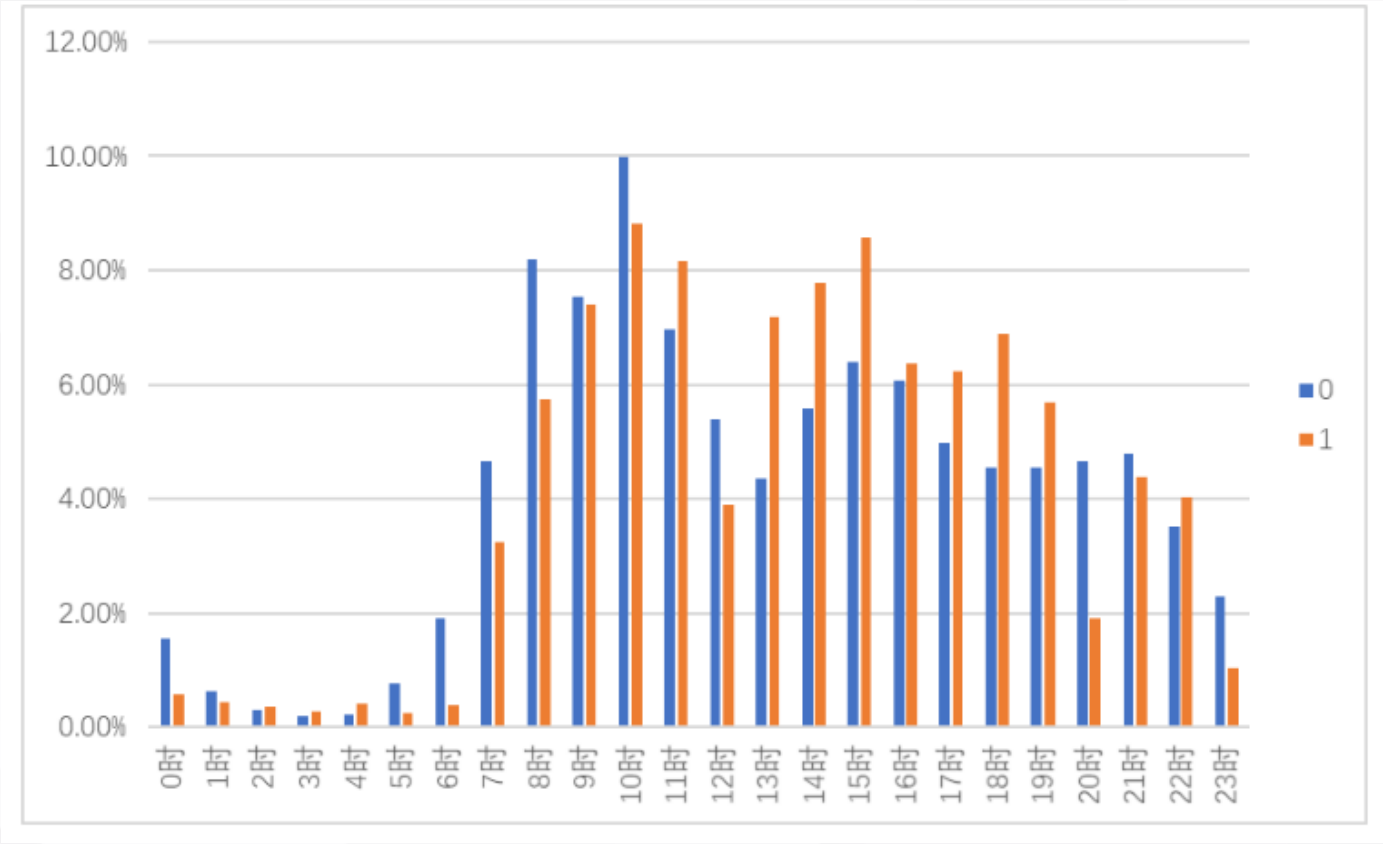
Day



日期字段
统计一个月里每一天在app上操作的用户的总数，可以看出，用户行为数据在day上呈现出了周期性的特征。每隔几天就会有一个高峰，这个日期的间隔大概就是一周的时间，实际上这是符合正常情况的，周末的时候手机app的频率会比较高

数据探索

Time

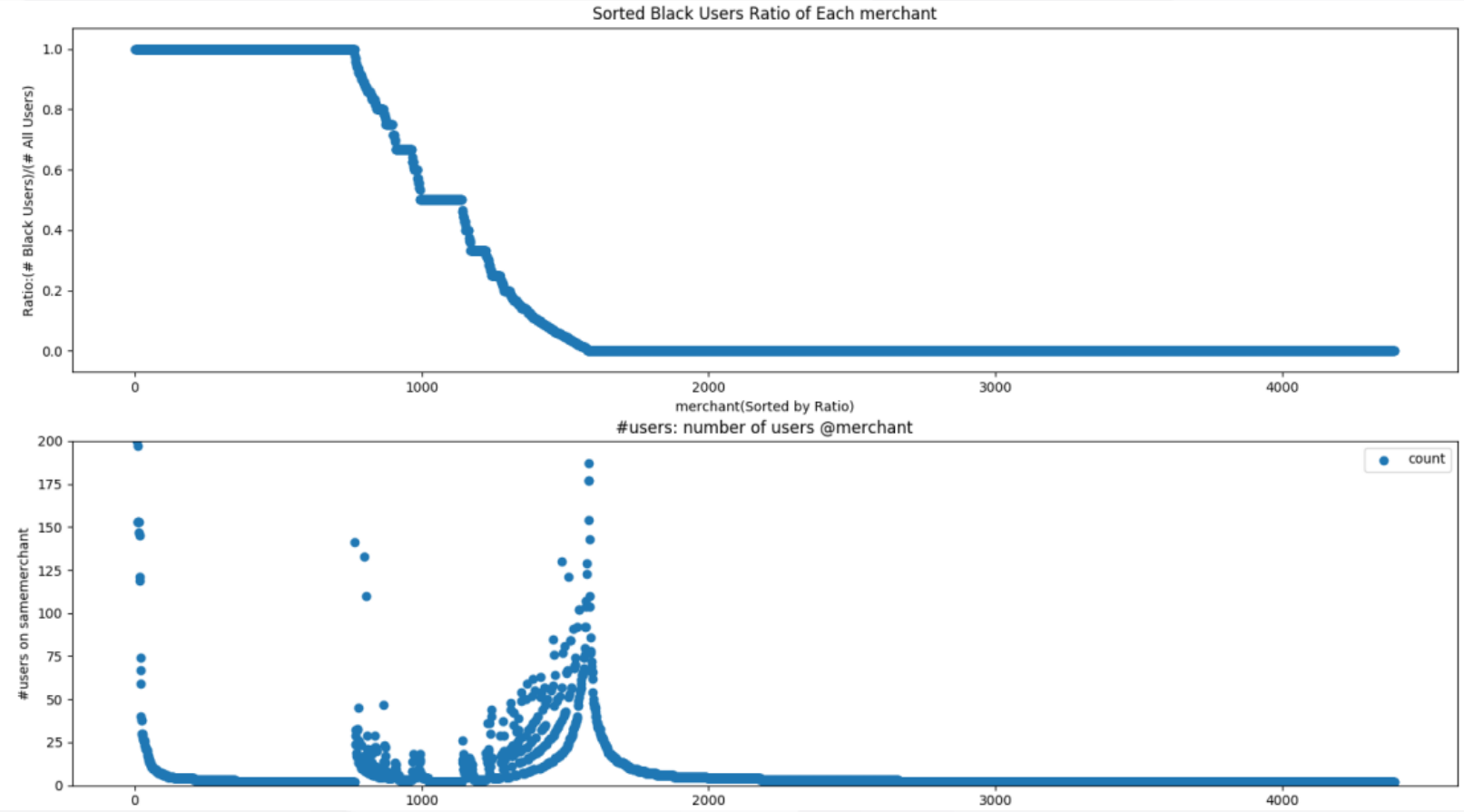


时间字段
统计一天之中每个时段在app上操作的用户的总数，可以看出，用户在0点到6点是比较少的，在6点到晚上11点的用户操作比较多。

某些用户喜欢在周一到周五使用app，在凌晨0点到6点使用app，值得怀疑

数据探索

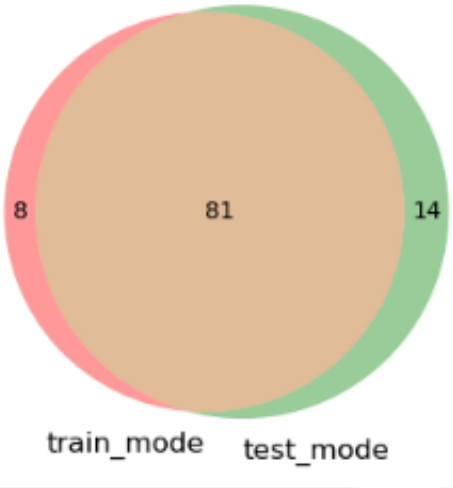
Merchant

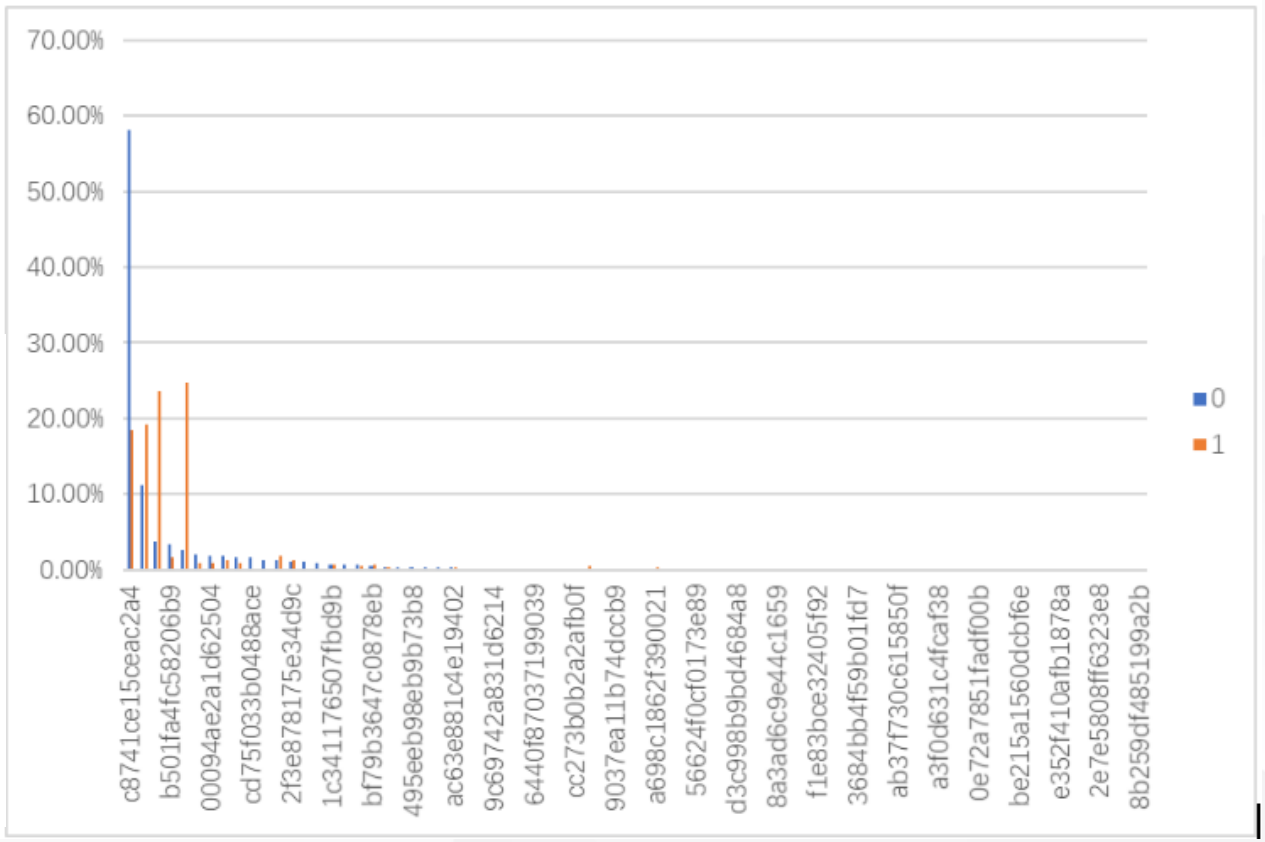


统计在每一个商家进行交易的用户的总数
观察每个商家的用户里面黑样本的比例
结合两幅图来看，在总数比较大的同时比例高，说明在此商店交易的用户有很大的嫌疑是羊毛党

数据探索

举例：对operation中正负样本在mode上的数据探索

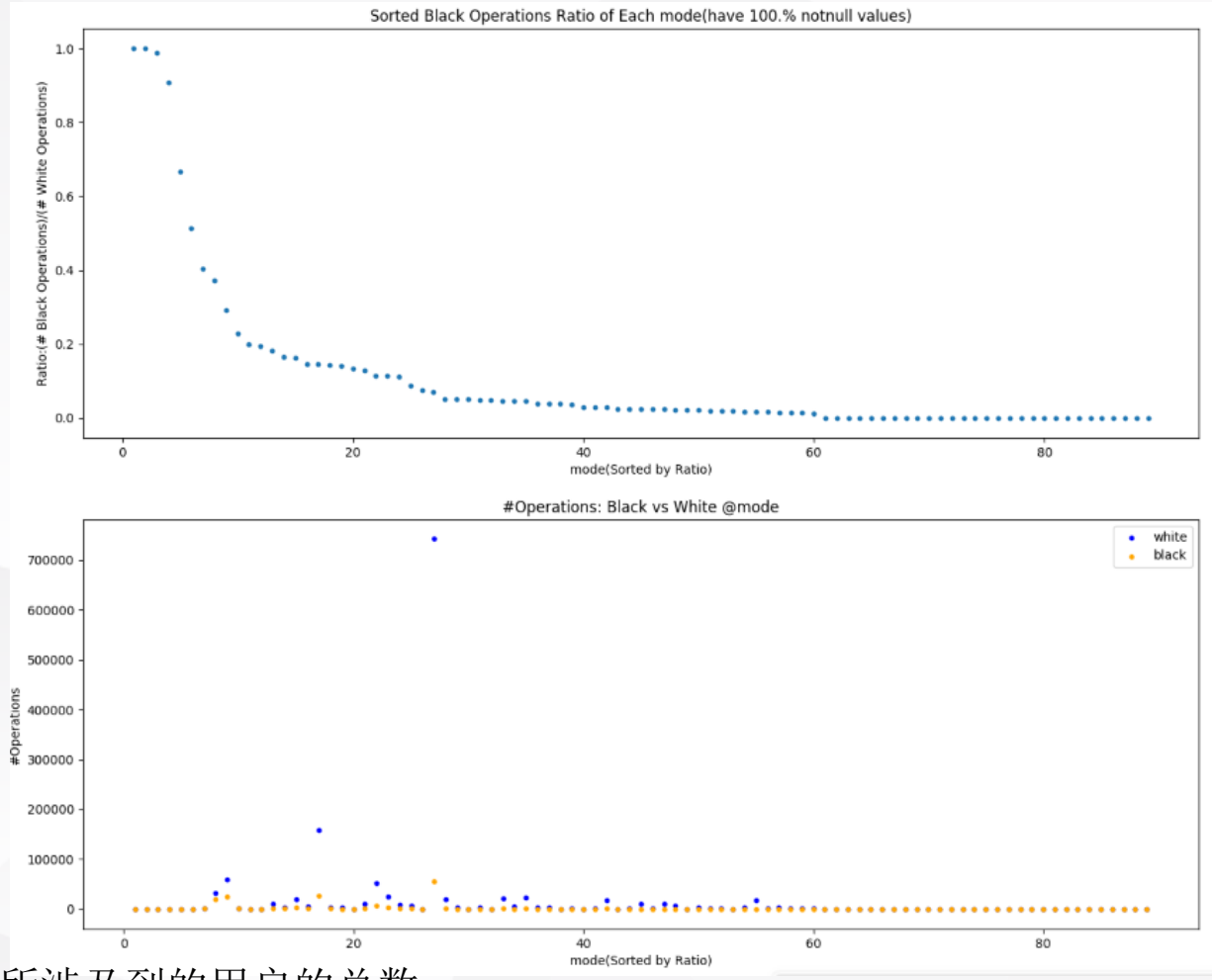




训练集里面的操作类型一共有89，测试集却有95种，对这种训练集有但是测试集没有的类型做特征提取不合理，应该做一个取交集的处理，对训练集、测试集共有的操作类型做提取

数据探索

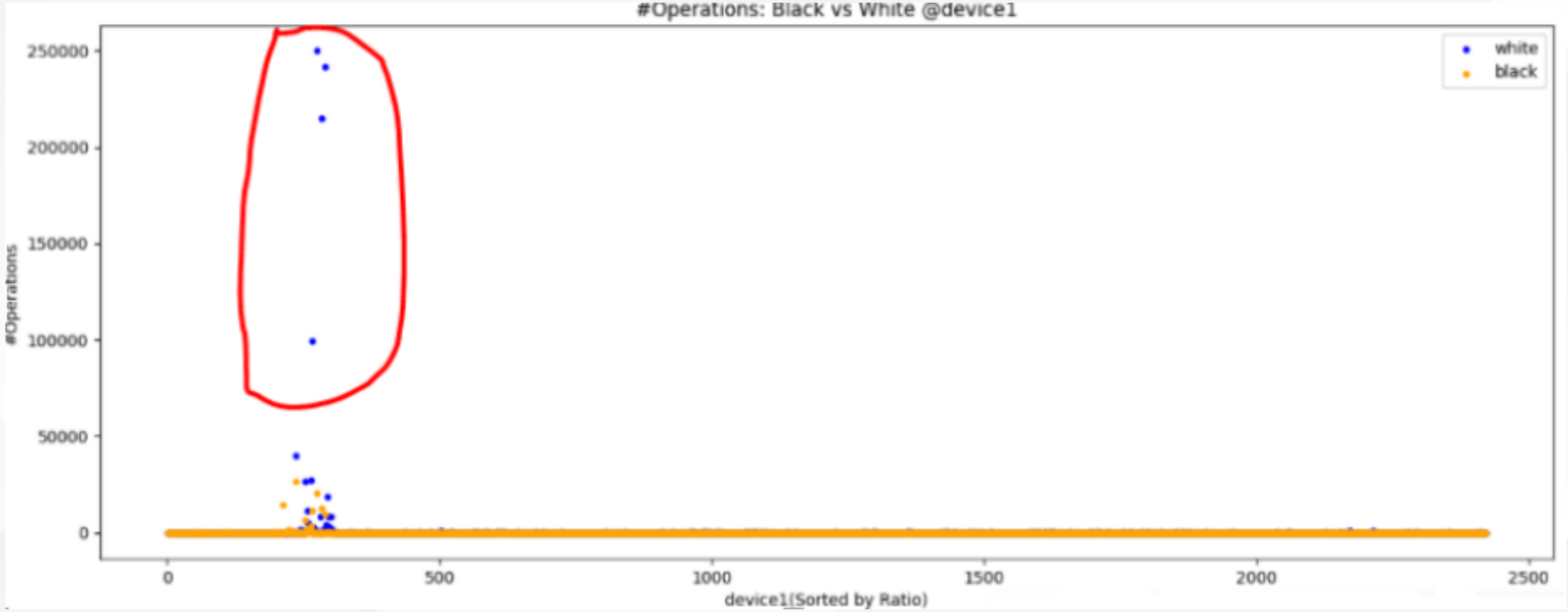
举例：对operation中正负样本在mode上的数据探索



统计每一种操作类型所涉及到的用户的总数
观察每个操作类型用户里面黑样本的比例
结合两幅图来看，在总数比较大的同时比例高，说明在此进行了某种操作的用户有很大的嫌疑是羊毛党

数据探索

异常数据处理



有部分设备编码，使用到它们的用户的总数达到了25万，而一般的设备编码使用人数都是这种较低的水平，我们将这样的记录视为异常。对于异常的数据，包括其他字段的类似异常，我们都是直接删除这些记录。

特征工程

频次统计

将类别字段，按用户id划分，然后统计每个类别的频次，都可以作为一种特征。
用户一个月里面进行的某一种操作的次数
用户一个月里面使用的某一种操作系统os的次数

os
mode
trans_type
…

Groupby用户id

① 最小值
② 最大值
③ 累加值
④ 平均数

	UID	os	os_count
0	10000	103	9
1	10001	102	49
2	10001	200	17
3	10001	104	1
4	10002	102	10
5	10002	101	1
6	10003	102	15
7	10004	102	34
8	10006	102	4
9	10006	200	3
10	10007	102	13

	UID	max	min	mean
0	10000	26	13	20.222222
1	10001	13	2	6.522388
2	10002	29	29	29.000000
3	10003	17	17	17.000000
4	10004	29	1	12.647059
5	10006	8	7	7.571429
6	10007	26	18	22.307692

频次统计

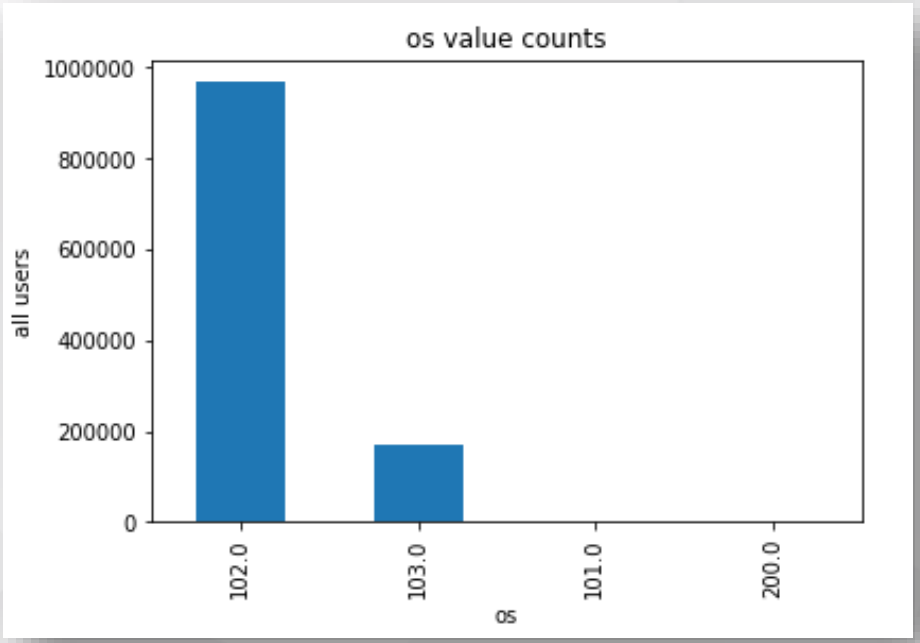
将类别字段每个取值频次都作为特征，将每个类别作为特征的列名，那么对应位置的数据就是对应类别的频次

op_Data.groupby('UID').os.value_counts().unstack().reset_index().fillna(0)

	UID	os	os_count
0	10000	103	9
1	10001	102	49
2	10001	200	17
3	10001	104	1
4	10002	102	10
5	10002	101	1
6	10003	102	15
7	10004	102	34
8	10006	102	4
9	10006	200	3
10	10007	102	13

os	UID	101	102	103	104	105	107	200
0	10000	NaN	NaN	9.0	NaN	NaN	NaN	NaN
1	10001	NaN	49.0	NaN	1.0	NaN	NaN	17.0
2	10002	1.0	10.0	NaN	NaN	NaN	NaN	NaN
3	10003	NaN	15.0	NaN	NaN	NaN	NaN	NaN
4	10004	NaN	34.0	NaN	NaN	NaN	NaN	NaN
5	10006	NaN	4.0	NaN	NaN	NaN	NaN	3.0
6	10007	NaN	13.0	NaN	NaN	NaN	NaN	NaN

频次统计

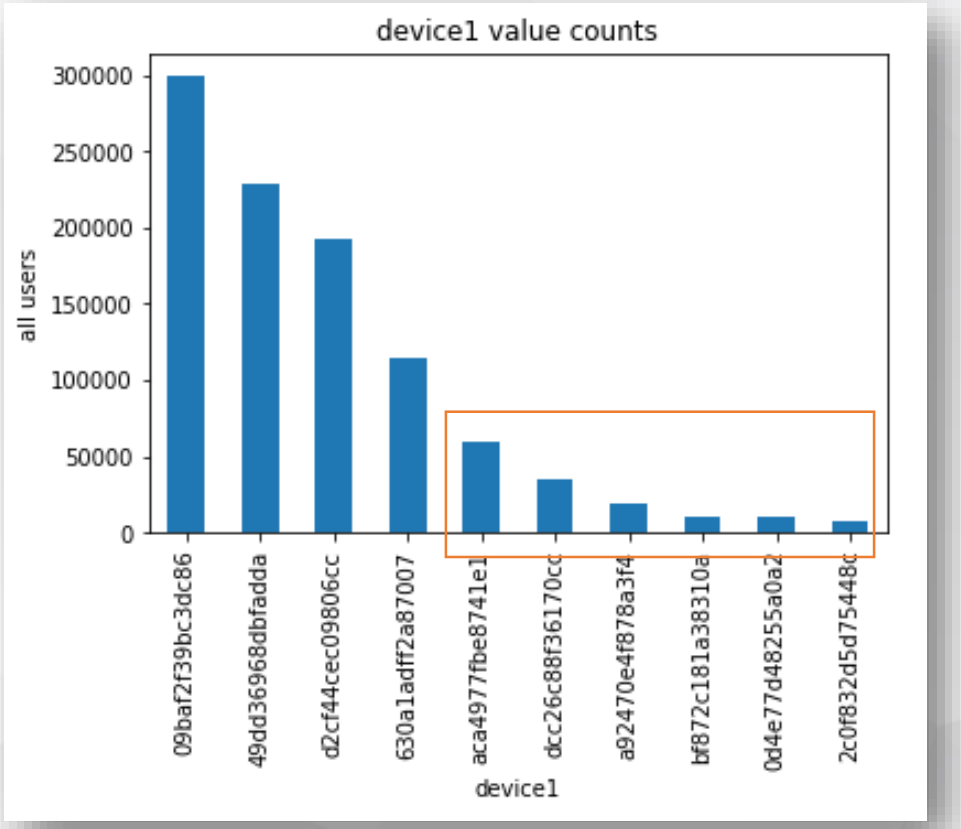


类别种类较少

类别值基本不变

直接按照刚才的方式就可以得到频次特征

频次统计



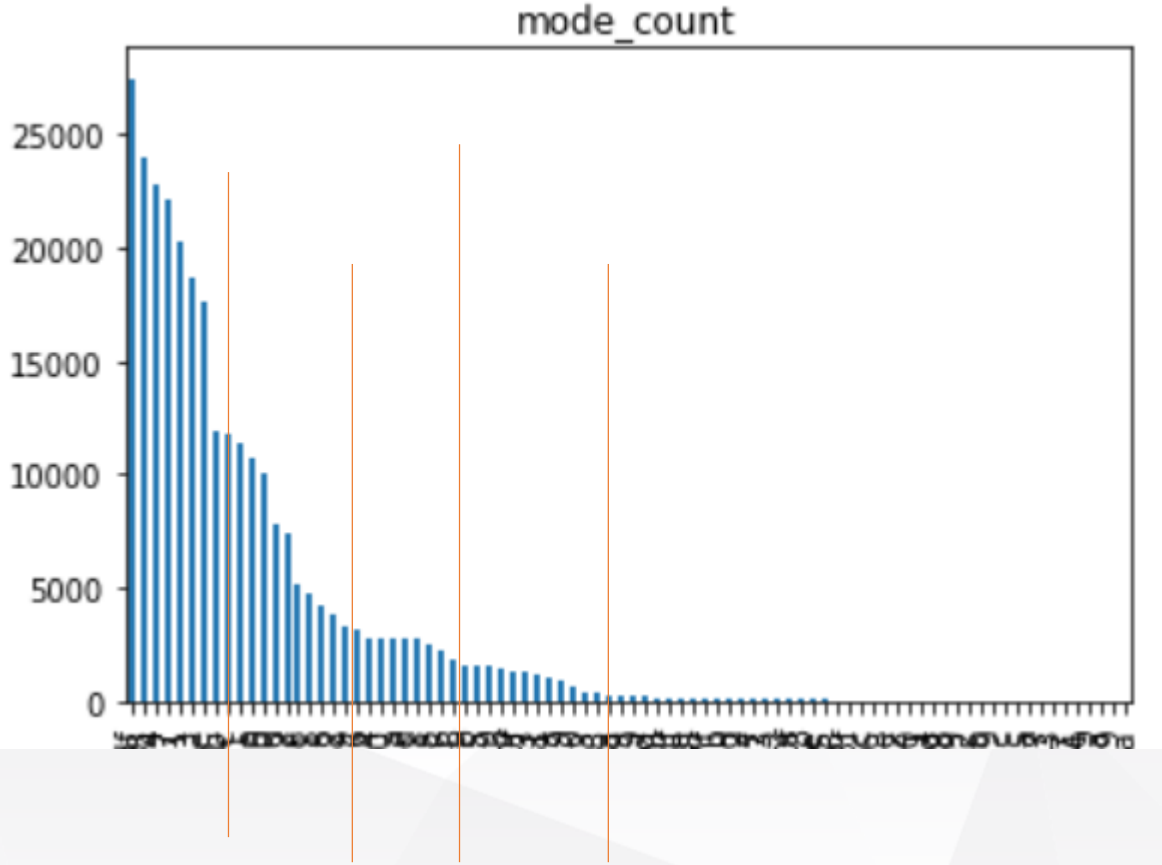
有一些类别字段的里面有很多类别

采取切割的方法，先统计每个类别在所有记录里面的频次，排序之后，就可得到这样的图。

频次较高的，我们保留原始类别，较低的，我们将它们归为 “其它类别”，也就是9类变成了5类。

之后再针对每个用户统计这5类的频次作为特征。

频次统计



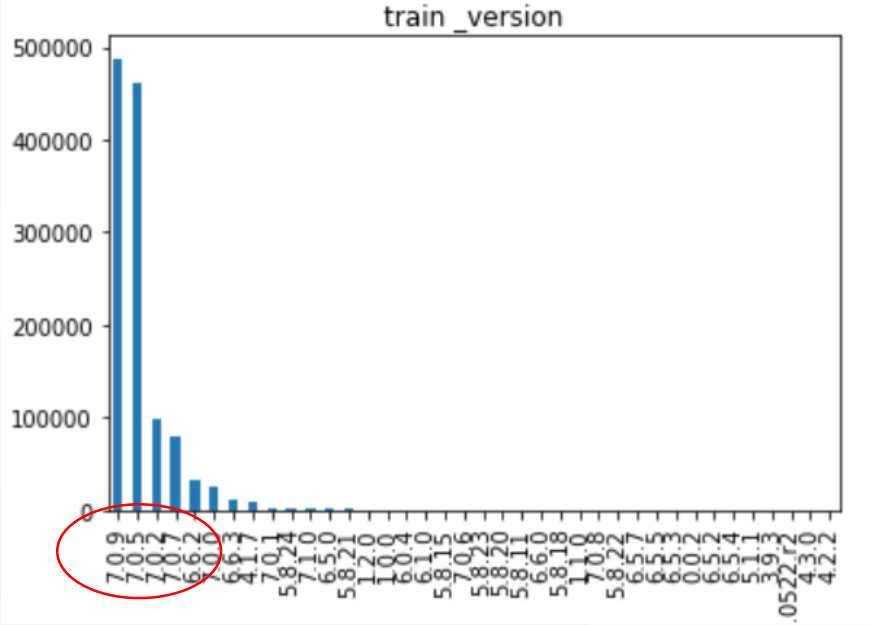
面对类别字段的里面有更多类别，

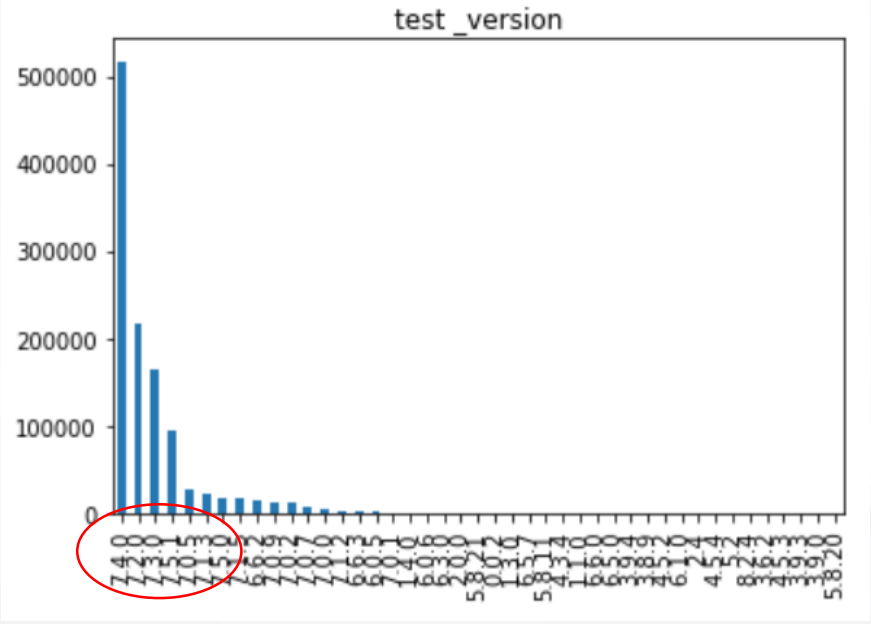
一样采取切割的方法，先统计每个类别在所有记录里面的频次，排序之后。

一样采取切割的方法，但这里不只是做一个划分，假设我们做k个划分，每个区间里的类别归为新的一个大类，就能得到k+1个新的类别。

流行度

频次统计还会遇到一个问题，比如说app的版本字段，训练集的用户多数使用7.0，而几个月之后的测试集的用户使用的多是7.4版本，我们按照刚才那一道直接统计频次并不合理。统计用户总数的时候发现它们的分布还是相似的，所以我们可以用这个数字作为一个替换，也就是“流行度”概念。





流行度

含义

字段映射为数值， 数值反映字段取值的用户数量

流行度替换了原始的版本号，类别字段就变成了数值型的字段，避开了训练集和测试集版本号差异大的问题

version	流行度
7.0.9	17432
7.0.5	16753
7.0.7	4004
7.0.2	3463
6.6.2	1331

Train

version	流行度
7.4.0	17521
7.2.0	14659
7.3.0	7819
7.5.1	4932
7.5.0	1657

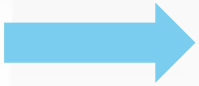
Test

地理信息字段处理

geohash编码：二维的经纬度转换成字符串

	geo_code
0	wskx
3	wm4v
5	ws90
6	ws45
7	ws2x
9	ws82
10	wkjm
11	wwdj
12	wmxb
13	wwew
14	wtgf

```
import geohash
lambda x:geohash.decode(x)
```



	geo_code
0	(25, 119)
3	(29, 105)
5	(25, 114)
6	(23, 115)
7	(25, 113)
9	(25, 113)
10	(23, 109)
11	(38, 115)
12	(31, 112)
13	(38, 118)
14	(33, 118)

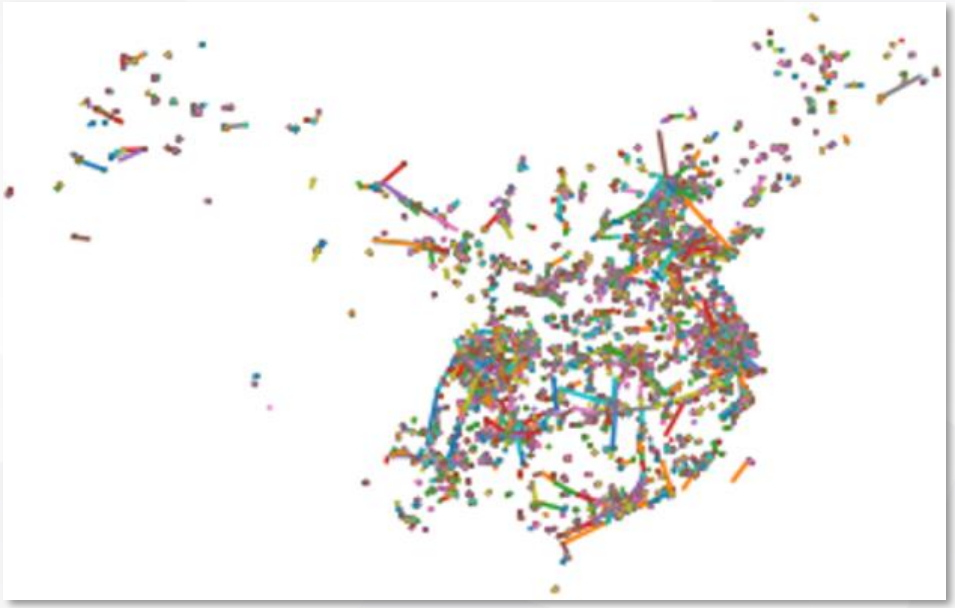
tips：一般的工具需要手动安装geohash包

python的话在cmd输入命令“pip install Geohash”

地理信息字段处理

“羊毛党”

- 恶意篡改地理位置信息
- 使用vpn



经纬度

按月统计　　按天统计

极值、跨度、方差等特征

活动范围、活动位置变化

特征工程-初步

按照类似于前面讲到的方法，对两个记录表里面的一些字段做了特征提取，可以得到下面这些特征，

- 1、4、6、7、10这些都是频次特征
- 第5条利用到的是缺失记录的信息
- 第9条利用到的是刚刚分析的日期字段、时间字段的信息

- 用户行为频次、交易频次、交易转化率；
- 用户发生行为时经纬度统计特征；
- 用户交易金额统计特征（mean，max，min，skew等）；
- 用户交易时使用的资金来源种类个数；
- 用户行为发生时设备/环境信息缺失程度；
- 用户使用的设备/环境的种类个数；
- 用户当前行为与之前行为（某时间间隔内）设备/环境变化次数；
- 用户当前行为是否为常用设备/环境；
- 用户行为/交易时间差统计；
- 用户某种行为到发生交易的时间间隔小于某阈值的频次；
- 用户不同设备的使用时间长度；

模型选择

- 我们当时选择的模型lgb，在Python上只需要设置参数就可以学习

```
lgb.LGBMClassifier (boosting_type='gbdt', num_leaves=200,
                    reg_alpha=3,reg_lambda=5,
                    max_depth=-1,n_estimators=5000,
                    objective='binary', subsample=0.9,
                    colsample_bytree=0.77,subsample_freq=1,
                    learning_rate=0.05,random_state=1000,
                    n_jobs=16, min_child_weight=4,
                    min_child_samples=5, min_split_gain=0)
```

- 模型参数设置：到网上查看python的官方文档，理解参数含义

模型选择

我们还可以选择
现在用的比较多的树模型

XGBoost
LightGBM
CatBoost
RandomForest
ExtraTree
Regularized Greedy Forest (RGF)
GradientBoostingClassifier

其他模型　　逻辑回归、baiveBayes、...

扩展思考

- 只用统计特征是不够的，因为测试数据和训练数据时间相隔太久，复赛测试数据与训练数据时间相隔更久。
- 前面所讲到的实际上是对羊毛党的行为模式的一个探索，而我们还可以考虑利用羊毛党之间的关联关系，比如说，羊毛党薅羊毛的时候，有没有可能很多个账号连接到同一个wifi、有没有可能很多个账号同时到同一个商家做同样的交易，我们可以考虑一下怎么把这样的潜在信息提取出来
- 对于关联信息这一点，我们当时前十的队伍都多多少少考虑到了**关联图谱、图数据库**两个技术，可以往这方面探索一下

磨刀功夫

注意记录：避免重复工作、避免遗漏信息

- 分析数据、业务——潜在的信息
- 特征改动——线上、线下效果
- 参数调整——线上、线下效果

自动日志：python的logging包

手动记录

磨刀功夫

多看评论和赛后解法

- 每一次比赛后其实都对实际的Data Mining问题有新的认识，总结自己的方案和其他top的方案，这是在比赛方面提高的关键

不断学习

- 不断接触前沿paper，不断学习当下最新算法，可以参考工业界有效的方案

要求

- 到Data Castle报名参赛，单人参赛。注意队名为：DM+学号
  http://www.dcjingsai.com/common/cmpt
- 下载数据，完成比赛
- 考核：线上分数+模型亮点
- baseline会发到微信群里