TalkingData AdTracking Fraud Detection Challenge

2018-12-03

目录

1. 问题描述

预测用户在点击 app 广告后下载 app 的概率。以 AUC 作为评判标准。

2. 数据字段介绍

字段	字段说明
ip	点击 ID 地址；
app	App_ID；
device	设备（苹果 6，苹果 7，华为 mate7 等）；
os	用户手机 OS 版本 ID；
channel	移动广告发布平台；
click_time	点击时间，天&时&分；
attributed_time	若用户下载了 app, 这就是下载时间；
is_attributed	是否下载，0 或 1；

3. 数据分析

3.1 数据量统计

Train_data	**Test_data**
包含：6-9 号数据；	包含：9-10 号数据；
数据量：184,903,891	数据量：57,537,506
正负样本：456846 : 184447045 = 1 : 403.7	NAN

Train_date	**Train_data**	**数据量占比**
2017-11-06	9,308,568	5.03427%
2017-11-07	59,633,310	32.251%
2017-11-08	62,945,075	34.042%
2017-11-09	53,016,937	28.6727%

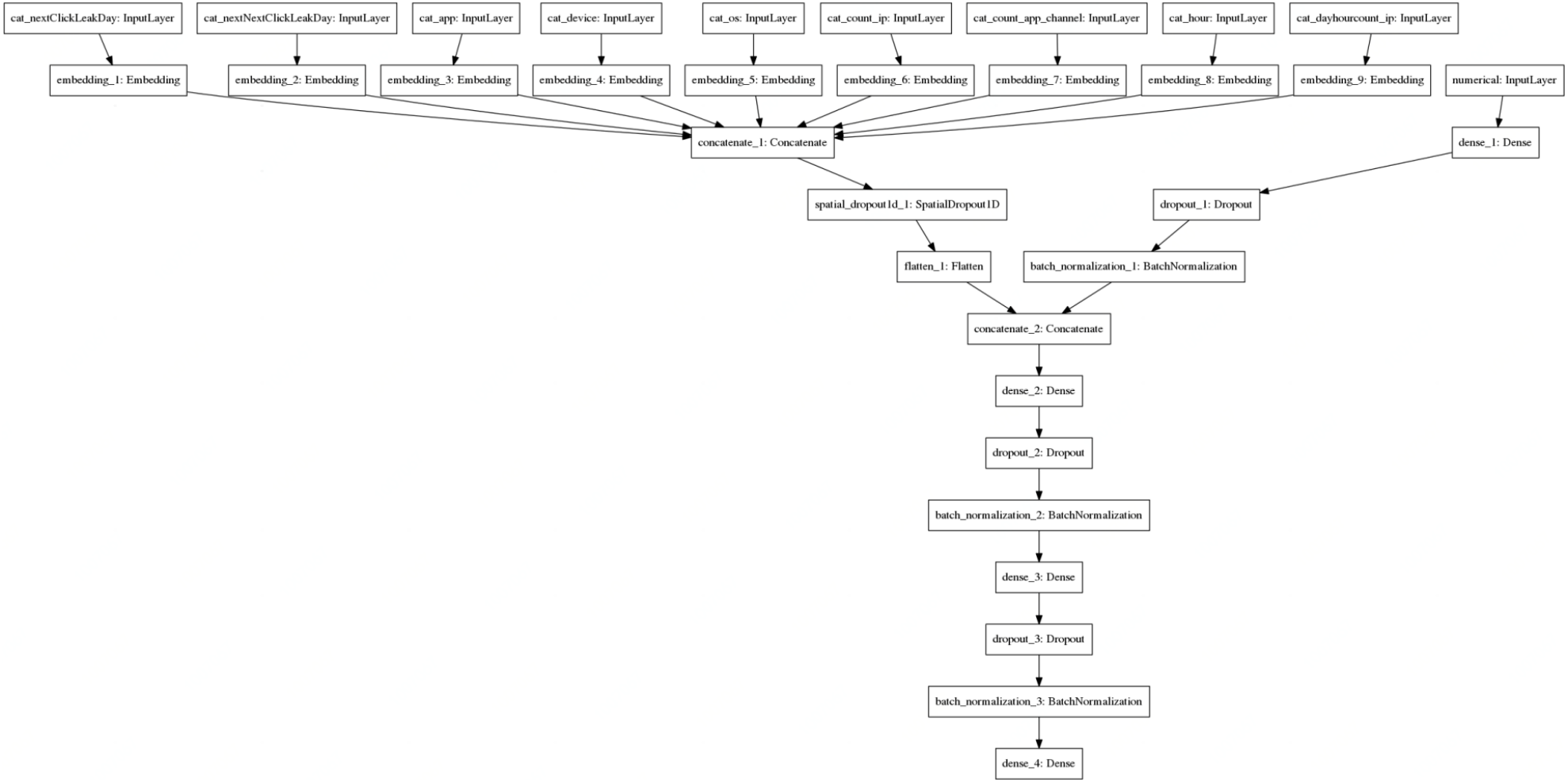
Test_date	Test_data	数据量占比
2017-11-09	9,802,613	17.0369%
2017-11-10	47,734,892	82.9631%

4. 第四名模型

采用 11.07 和 11.08 两天的数据作为训练集，在 11.09 的 4:00-14:00 数据上进行验证预测；最后以 11.07-11.09 三天的数据作为训练集，再次训练来预测 11.10 的数据。

（1）model1: lightgbm；
（2）model2: 网络模型；
（3）stacking(model1, model2)

model2（网络模型）结构：



5. 第四名特征（20+4+3+10+1+1+24+123=186 维）

第一部分：统计特征（12+3+5=20 维）

1. count 特征（12 维）

（1）根据 click_time 字段，分离出 day / hour；

（2）统计点击次数：groupby[X]['is_attributed'].count()；

X 如下：

字段组合	解释说明
'app_channel'	以 app、channel 分组，计数；
'app_device_channel_day_hour'	以 app、device、channel、day、hour 分组，计数；
'app_device_day_hour'	以 app、device、day、hour 分组，计数；
'app_os_channel_day_hour'	以 app、os、channel、day、hour 分组，计数；
'ip_day'	以 ip、day 分组，计数；
'ip'	以 ip 分组，计数；
'ip_app_device_channel_day'	以 ip、app、device、channel、day 分组，计数；
'ip_app_device_day'	以 ip、app、device、day 分组，计数；
'ip_app_device_os_day_hour'	以 ip、app、device、os、day、hour 分组，计数；
'ip_app_os_channel'	以 ip、app、os、channel 分组，计数；
'ip_app_os_channel_day'	以 ip、app、os、channel、day 分组，计数；
'ip_os'	以 ip、os 分组，计数；

2. 有关时间 count 特征（3 维）

（1）根据 click_time 字段，分离出 day / hour / min；

（2）根据 min 字段，制造整十分钟 min10 字段（即：min 对 10 取整）；

（3）根据 hour 和 min / min10，制造小时分钟 hourmin / hourmin10 字段；

（4）统计点击次数：groupby[X]['is_attributed'].count()；

X 如下：

字段组合	解释说明
'app_day_hourminute'	以 app、day、hourminute 分组，计数；
'device_os_day_hourminute10'	以 device、os、day、hourminute10 分组，计数；
'ip_device_os_day_hourminute10'	以 ip、device、os、day、hourminute10 分组，计数；

3. Ratio 统计特征（5 维）

（1）根据 device 和 os 字段，制造 machine 字段；

（2）统计占比：

groupby[X]['is_attributed'].count() / groupby[X.split('_')[0]]['is_attributed'].count()；

X 如下：

字段组合	解释说明
'ip_machine'	以 ip、machine 分组计数占以 ip 分组计数的比例；
'ip_channel'	以 ip、channel 分组计数占以 ip 分组计数的比例；

'machine_ip'	以 machine、ip 分组计数占以 machine 分组计数的比例；
'app_channel'	以 app、channel 分组计数占以 app 分组计数的比例；
'channel_app'	以 channel、app 分组计数占以 channel 分组计数的比例；

第二部分：统计累积特征（2+1+1=4 维）

1. 原始时间按照 click_time 字段升序，统计累积特征；（2 维）

（1）根据 click_time 字段，分离出 day / hour；

（2）统计累积特征：groupby(X).cumcount()；

X 如下：

字段组合	解释说明
'ip_app_device_os_day_hour'	按照 click_time 字段升序， 以 ip、app、device、os、day、hour 分组，累积计数；
'ip_day'	按照 click_time 字段升序，以 ip、day 分组，累积计数；

2. 原始时间按照 click_time 降序，统计累积特征；（1 维）

（1）根据 click_time 字段，分离出 day / hour；

（2）统计累积特征：groupby(X).cumcount()；

X 如下：

字段组合	解释说明
'app_device_os_day'	按照 click_time 字段降序， 以 app、device、os、day 分组，累积计数；

3. 统计累积占比特征；（1 维）

（1）根据 click_time 字段，分离出 day / hour；

（2）累积特征与计数特征比值：df[cumcount (X)] / (df[count(X)]-1)；

X 如下：

字段组合	解释说明
'ip_day'	按照 click_time 字段升序， 以 ip、day 分组的累积计数占以 ip、day 分组计数的比例；

第三部分：点击时间差特征（time-delta）（2+1=3 维）

1. 下次（下下次）距上次点击的时间差特征；（2 维）

（1）根据 click_time 字段，分离出 day；

（2）计算下次距上次点击时间差（单位：s）：

(df.groupby(X).click_time.shift(-1) - df.click_time + 1).fillna(999999)

（3）计算下下次距上次点击时间差（单位：s）：

(df.groupby(X).click_time.shift(-2) - df.click_time + 1).fillna(999999)

X 如下：

字段组合	解释说明
'day_ip_app_device_os'	按照 click_time 字段升序， （1）以 day、ip、app、device、os 分组的下次点击时间与上次点击时间差； （2）以 day、ip、app、device、os 分组的下下次点击时间与上次点击时间差；

2. 转化时间差特征，生成类别新特征（nextClickLeakDayFlt）；（1 维）

（1）下次与上次点击时间间隔<30s，新字段特征为 0；

（2）下次与上次点击时间间隔[30, 1800s]，新字段特征为 1；

（3）否则，新字段特征为 2；

第四部分：时间 unique 计数特征（1+1+8=10 维）

1. 统计 unique(day)特征；(1 维)

（1）根据 click_time 字段，分离出 day；

（2）统计活跃的天数量：

```
df[[X, 'day']].groupby(by=X)[ 'day'].nunique()
```

X 如下：

字段组合	解释说明
'ip'	以 ip 分组，统计 unique(day)数；

2. 统计 unique(day_hour)特征；(1 维)

（1）根据 click_time 字段，分离出 day / hour；

（2）统计活跃的天&小时数量：

```
df[[X, 'day', 'hour']].groupby(by=X)[['day', 'hour']].nunique()
```

X 如下：

字段组合	解释说明
'ip'	以 ip 分组，统计 unique([day, hour])数；

3. 统计 unique(day_hour_min)特征；（4*2=8 维）

（1）根据 click_time 字段，分离出 day / hour / min；

（2）根据 min 字段，制造出整十分钟 min10 字段；

（3）根据 hour 和 min/min10 字段，制造出小时分钟 hourmin / hourmin10 字段；

（4）根据 day 和 hourmin / hourmin10 字段，制造出天小时分钟 dayhourmin / dayhourmin10 字段；

（5）统计活跃的 dayhourmin / dayhourmin10 数量：

```
df[[X, Y]].groupby(by=X)[[Y]].nunique()
```

X、Y 如下：

X 字段组合	Y 字段组合	解释说明
'ip'	'dayhourminute'	以 ip 分组，统计 unique(dayhourminute)数；
'app_os_channel'	'dayhourminute'	以 app、os、channel 分组，统计 unique (dayhourminute)数；
'ip_channel'	'dayhourminute'	以 ip、channel 分组，统计 unique (dayhourminute)数；
'ip_device_os'	'dayhourminute'	以 ip、device、os 分组，统计 unique (dayhourminute)数；
'ip'	'dayhourminute10'	以 ip 分组，统计 unique(dayhourminute10)数；
'app_os_channel'	'dayhourminute10'	以 app、os、channel 分组，统计 unique (dayhourminute10)数；
'ip_channel'	'dayhourminute10'	以 ip、channel 分组，统计 unique (dayhourminute10)数；
'ip_device_os'	'dayhourminute10'	以 ip、device、os 分组，统计 unique (dayhourminute10)数；

第五部分：方差特征（1 维）

1. 方差特征；（1 维）

（1）根据 click_time 字段，分离出 day / hour；

（2）计算方差：

groupby(by=X[0:len(X)-1])[X[len(X)-1]].var()

X 如下：

字段组合	解释说明
'ip_device_hour'	以 ip、device 分组，计算 hour 方差；

第六部分：common_ip 特征（1 维）

1. 统计 common_ip 特征；（1 维）

（1）根据日期，筛选每天均有访问的 ip 的日志数据；

（2）统计 common_ip 数据中的 ip 每天的访问次数；

（3）进而计算 ip 每天访问次数的均值与方差；

（4）进而计算方差与均值的比值关系（True / False）；

第七部分：ratio 特征（8*3=24 维）

1. 统计 ratio 特征；（8 维）

（1）统计 unique 数据：df[X].groupby(X.split('_')[:-1])[X.split('_')[-1]].nunique()；

（2）统计 count 数据：df[X].groupby(X.split('_')[:-1])[X.split('_')[-1]].count()；

（3）计算占比特征：ratio = nunique() / count()；

X 如下：

字段组合	解释说明
'day_ip_machine'	以 day、ip 分组的 unique(machine)占以 day、ip 分组的 count(machine)比例；
'day_ip_os'	以 day、ip 分组的 unique(os)占以 day、ip 分组的 count(os)比例；
'day_ip_device'	以 day、ip 分组的 unique(device)占以 day、ip 分组的 count(device)比例；
'day_ip_app'	以 day、ip 分组的 unique(app)占以 day、ip 分组的 count(app)比例；
'day_ip_channel'	以 day、ip 分组的 unique(channel)占以 day、ip 分组的 count(channel)比例；
'machine_app'	以 machine 分组的 unique(app)占以 machine 分组的 count(app)比例；
'machine_channel'	以 machine 分组的 unique(channel)占以 machine 分组的 count(channel)比例；
'machine_ip'	以 machine 分组的 unique(ip)占以 machine 分组的 count(ip)比例；

第八部分：count, sum 和自定义 log 特征（3*41=123 维）

1. 统计 count, sum, 自定义 log 特征；

（1）根据 click_time 字段，分离出 day / hour；

（2）筛选 Train_data 数据，保留 hour in [12, 22]的数据；

（3）根据 day 字段，选择其他 days 数据，统计：

groupby(by=X)[['is_attributed']].agg(['count', 'sum'])、

np.log((df['sum']/pos)/((df['count']-df['sum']+0.1**8)/neg)+1) 特征，

代替当前 day 数据的 count、sum、自定义 log 特征；

（4）筛选 Test_data 数据，保留 hour in [12, 22] 的数据；

（5）根据 day 字段，统计当前 day 数据的 count、sum、和自定义 log 特征；

X 表示如下：

字段组合	解释说明 1 (count)	解释说明 2 (sum)	解释说明 3 (log)
'ip'	以 ip 分组，统计点击次数；	统计下载次数；	根据 count 和 sum，计算 log 值；
'app'	以 app 分组，统计点击次数；	同上；	同上；
'device'	以 device 分组，统计点击次数；	同上；	同上；
'os'	以 os 分组，统计点击次数；	同上；	同上；
'channel'	以 channel 分组，统计点击次数；	同上；	同上；
'ip_app'	以 ip、app 分组，统计点击次数；	同上；	同上；
'ip_device'	以 ip、device 分组，统计点击次数；	同上；	同上；
'ip_os'	以 ip、os 分组，统计点击次数；	同上；	同上；
'ip_channel'	以 ip、channel 分组，统计点击次数；	同上；	同上；
'app_device'	以 app、device 分组，统计点击次数；	同上；	同上；
'app_os'	以 app、os 分组，统计点击次数；	同上；	同上；
'app_channel'	以 app、channel 分组，统计点击次数；	同上；	同上；
'ip_app_device'	以 ip、app、device 分组，统计点击次数；	同上；	同上；
'ip_app_os'	以 ip、app、os 分组，统计点击次数；	同上；	同上；
'ip_app_channel'	以 ip、app、channel 分组，统计点击次数；	同上；	同上；

'ip_device_os'	以 ip、device、os 分组，统计点击次数；	同上；	同上；
'ip_device_channel'	以 ip、device、channel 分组，统计点击次数；	同上；	同上；
'ip_os_channel'	以 ip、os、channel 分组，统计点击次数；	同上；	同上；
'app_device_os'	以 app、device、os 分组，统计点击次数；	同上；	同上；
'app_device_channel'	以 app、device、channel 分组，统计点击次数；	同上；	同上；
'app_os_channel'	以 app、os、channel 分组，统计点击次数；	同上；	同上；
'ip_app_device_os'	以 ip、app、device、os 分组，统计点击次数；	同上；	同上；
'ip_app_device_channel'	以 ip、app、device、channel 分组，统计点击次数；	同上；	同上；
'ip_app_os_channel'	以 ip、app、os、channel 分组，统计点击次数；	同上；	同上；
'ip_device_os_channel'	以 ip、device、os、channe 分组，统计点击次数；	同上；	同上；
'app_device_os_channel'	以 app、device、os、channel 分组，统计点击次数；	同上；	同上；
'ip_nextClickLeakDayFlt'	以 ip、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'app_nextClickLeakDayFlt'	以 app、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'device_nextClickLeakDayFlt'	以 device、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'os_nextClickLeakDayFlt'	以 os、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'channel_nextClickLeakDayFlt'	以 channel、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'ip_app_nextClickLeakDayFlt'	以 ip、app、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'ip_device_nextClickLeakDayFlt'	以 ip、device、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'ip_os_nextClickLeakDayFlt'	以 ip、os、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'ip_channel_nextClickLeakDayFlt'	以 ip、channel、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'app_device_nextClickLeakDayFlt'	以 app、device、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'app_os_nextClickLeakDayFlt'	以 app、os、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'app_channel_nextClickLeakDayFlt'	以 app、channel、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'device_os_nextClickLeakDayFlt'	以 device、os、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'device_channel_nextClickLeakDayFlt'	以 device、channel、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；
'os_channel_nextClickLeakDayFlt'	以 os、channel、nextClickLeakDayFlt 分组，统计点击次数；	同上；	同上；

6. 第四名特征重要度

Count 重要度（top12）

Importance (count)	**feature**	**特征说明**
6728	cat_channel	ID 特征；移动广告发布平台；
5751	dayhourcount_ip	unique 计数特征；以 ip 分组，统计 unique([day, hour])数；
4234	cat_hour	ID 特征；小时；
4162	cat_app	ID 特征；app_id；
3959	cat_os	ID 特征；用户手机 OS 版本 id；
2378	nextClickLeakDay	时间差特征；下次点击距离上次点击的时间差；
1464	cumratio_ip_day	累积占比特征；以 ip 和 day 分组的累积计数特征与计数特征比值；
1378	recumcount_app_device_os_day	累积特征；按照 click_time 字段降序，以 app、device、os、day 分组，累积计数；
1334	uniqueCount_day_ip_machine	unique 计数特征；以 day、ip 分组的 unique(machine)；
1305	var_ip_device_hour	方差特征；以 ip、device 分组，计算 hour 方差；
1231	count_ip_device_os_day_hourminute10	计数特征；以 ip、 device、os、day、hourminute10 分组，统计点击次数；
1231	count_ip	计数特征；以 ip 分组，统计点击次数；

Gain 重要度（top10）

Importance (gain)	**feature**	**特征说明**
34127526.623	WOEBnd_app_channel_nextClickLeakDayFlt	自定义 log 特征；
17903097.636	WOEBnd_app_os_channel	自定义 log 特征；
13450630.453	WOEBnd_app_os_nextClickLeakDayFlt	自定义 log 特征；
7521631.315	WOEBnd_app_device_os_channel	自定义 log 特征；
5192578.811	WOEBnd_app_nextClickLeakDayFlt	自定义 log 特征；
4216560.650	uniqueCount_day_ip_app	unique 计数特征；以 day、ip 分组的 unique(app)；
2872146.215	WOEBnd_app_channel	自定义 log 特征；
2768569.848	nextClickLeakDay	时间差特征；下次点击距离上次点击的时间差；
2552346.780	WOEBnd_app_device_os	自定义 log 特征；
2257666.532	cat_app	ID 特征；app_id；

7. Top1-Top3 特征

Top1:

1. 5 个原始分类特征（ip, os, app, channel, device），groupby().count()；
   1.1 接下来 1 小时和 6 小时的点击数 (count)；
   1.2 计算前向和后向的 click 时间差特征（time-delta）；
   1.3 历史点击的平均下载率；

2. 对分类变量的组合(共 20 种)，尝试用 LDA / NMF / LSA（共 3 种）得到嵌入(embedding)特征；
   2.1 设置 n_component=5，得到 20*5*3=300 维特征；

Top2:

1. 5 个原始分类特征（ip, os, app, channel, device）；
   1.1 groupby().count()；
   1.2 groupby().cumcount()；
   1.3 groupby().nunique()；

2. 时间差特征(time-delta)；

3. 计算每个 ip 在某些 app / os / channel 上的点击数(选取点击频率最高的几个)；

Top3:

1. 时间差特征(time-delta)；
   1.1 每个点击的前 5 次与后 5 次点击之间的时间差；

2. 5 个原始分类特征（ip, os, app, channel, device）+时间（day, hour）；
   1.1 groupby().count()；
   1.2 groupby().cumcount()；
   1.3 groupby().nunique()；
   1.4 groupby().mean();
   1.5 groupby().var();

8. 参考资料

https://github.com/CuteChibiko/TalkingData
https://zhuanlan.zhihu.com/p/36852456