TalkingData AdTracking Fraud Detection Challenge

2018-12-03

**目录**

[1. 问题描述 - 1 -](#_Toc531693168)

[2. 数据字段介绍 - 1 -](#_Toc531693169)

[3. 数据分析 - 1 -](#_Toc531693170)

[3.1数据量统计 - 1 -](#_Toc531693171)

[4. 第四名模型 - 1 -](#_Toc531693172)

[5. 第四名特征（20+4+3+10+1+1+24+123=186维） - 3 -](#_Toc531693173)

[第一部分：统计特征（12+3+5=20维） - 3 -](#_Toc531693174)

[第二部分：统计累积特征（2+1+1=4维） - 4 -](#_Toc531693175)

[第三部分：点击时间差特征（time-delta）（2+1=3维） - 5 -](#_Toc531693176)

[第四部分：时间unique计数特征（1+1+8=10维） - 6 -](#_Toc531693177)

[第五部分：方差特征（1维） - 7 -](#_Toc531693178)

[第六部分：common\_ip特征（1维） - 7 -](#_Toc531693179)

[第七部分：ratio特征（8\*3=24维） - 7 -](#_Toc531693180)

[第八部分：count, sum 和自定义log特征（3\*41=123维） - 8 -](#_Toc531693181)

[6. 第四名特征重要度 - 10 -](#_Toc531693182)

[7. Top1-Top3特征 - 11 -](#_Toc531693183)

[8. 参考资料 - 11 -](#_Toc531693184)

# 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。

# 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； |

# 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% |

# 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；

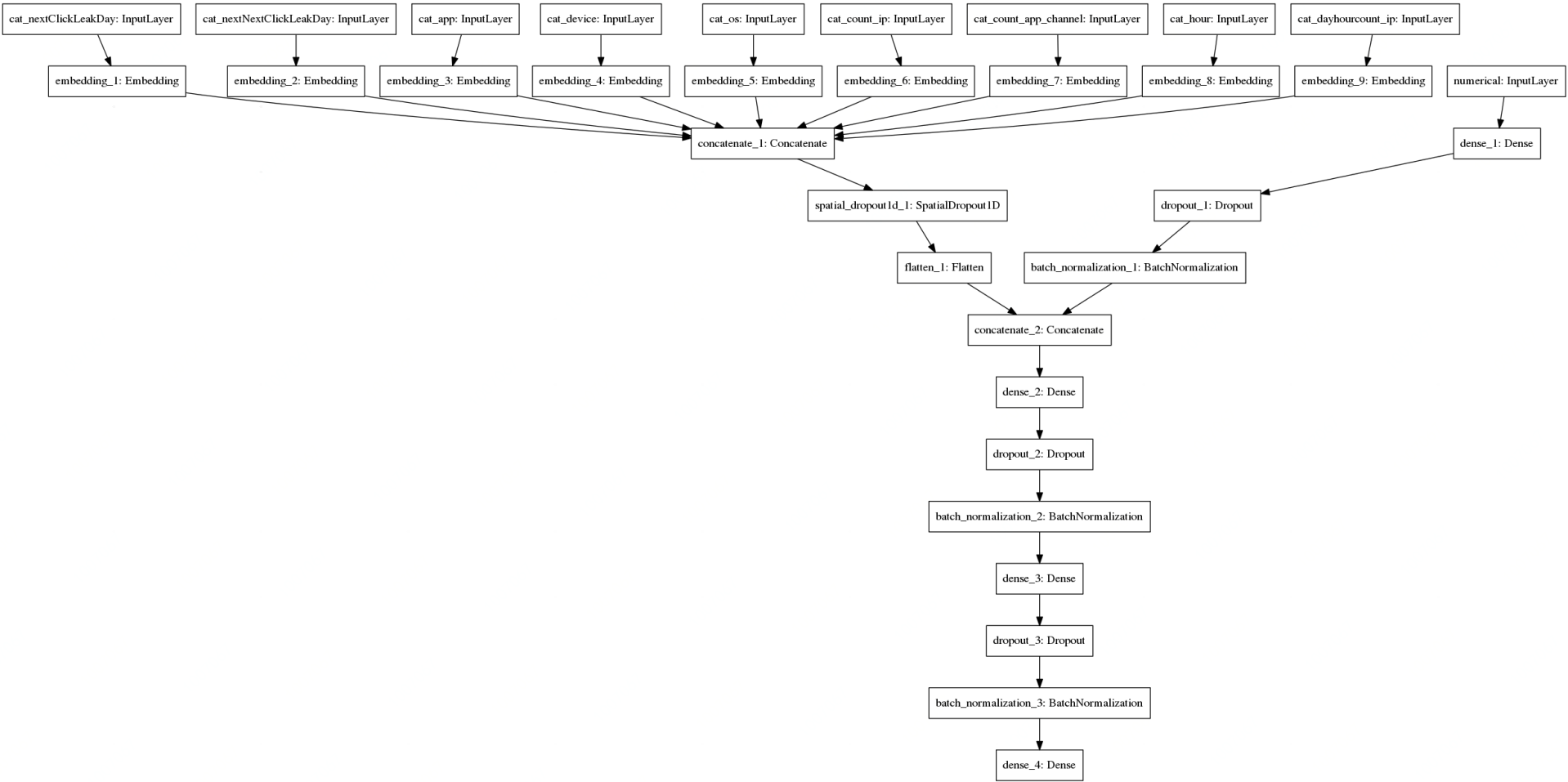
最后以11.07-11.09三天的数据作为训练集，再次训练来预测11.10的数据。

（1）model1: lightgbm；

（2）model2: 网络模型；

（3）stacking(model1, model2)

model2（网络模型）结构：



# 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； |

1. 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； |

1. 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； |

1. 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； |

1. 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； |

1. 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； |

1. 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； |

# 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； |

# 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();

# 参考资料

<https://github.com/CuteChibiko/TalkingData>

<https://zhuanlan.zhihu.com/p/36852456>