风险罗盘

17组 风险罗盘B组

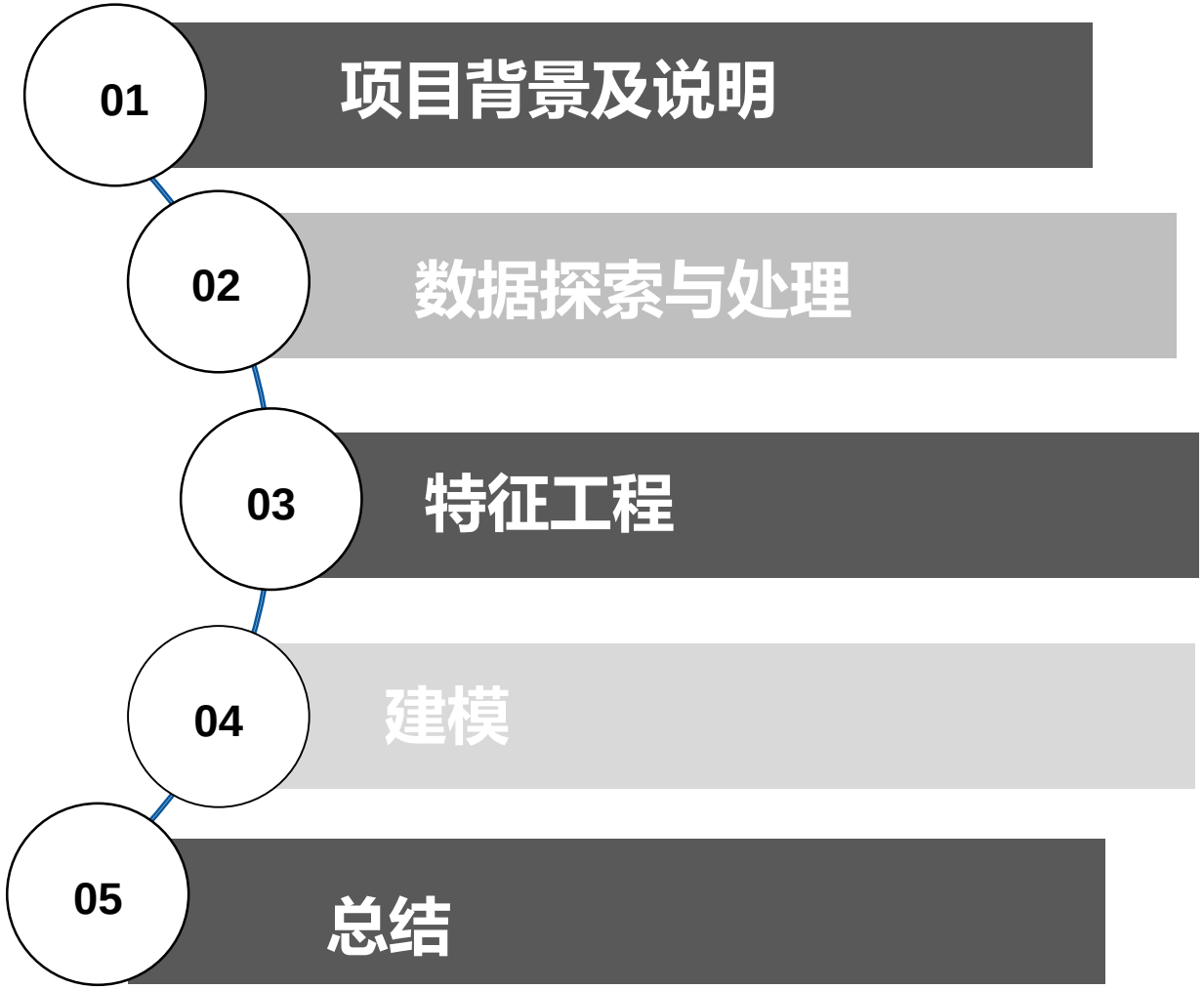
组员：顾胜达 孔文雁 刘田雨 李世鲲 潘俊廷 杨阁 杨梓溪 郑健

汇报人： 李世鲲 郑健

风险罗盘

2019 / 08 / 14

概述

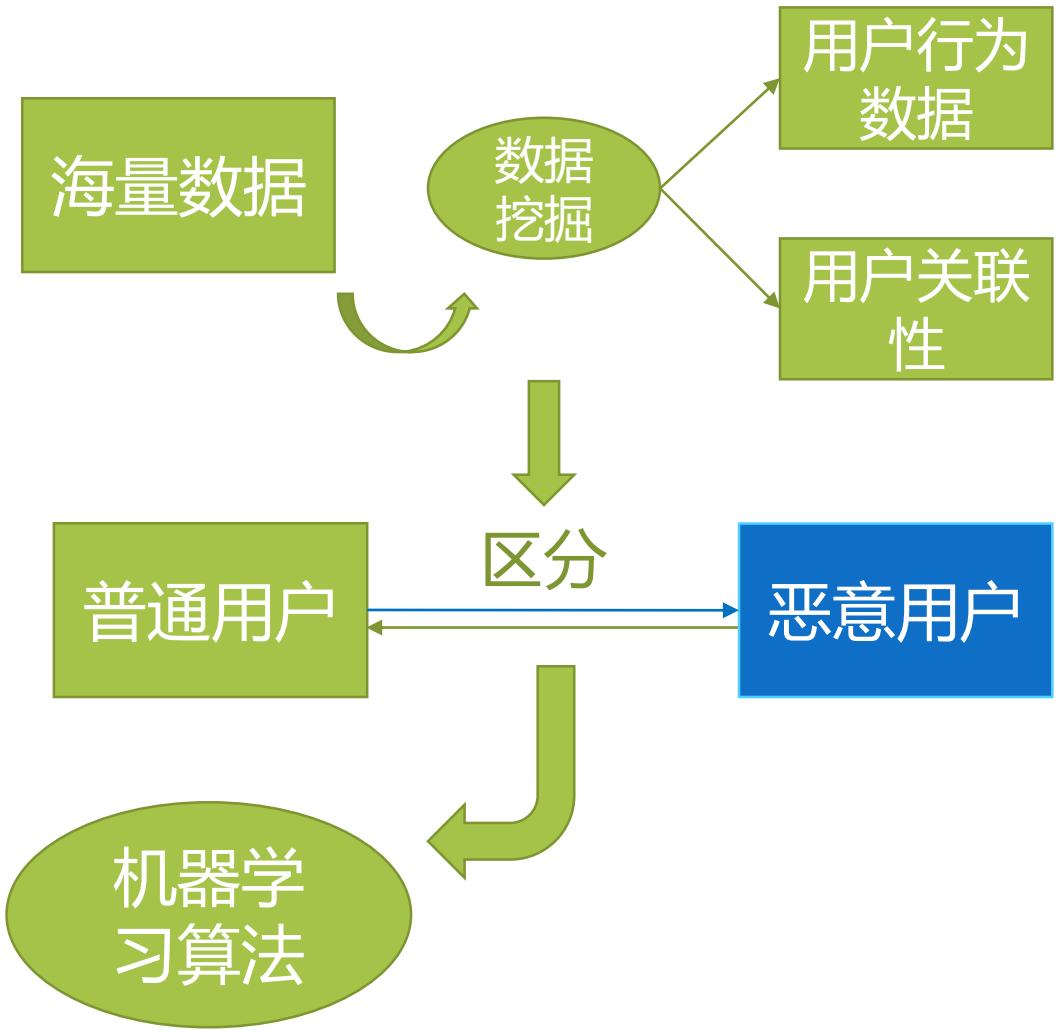


项目主题

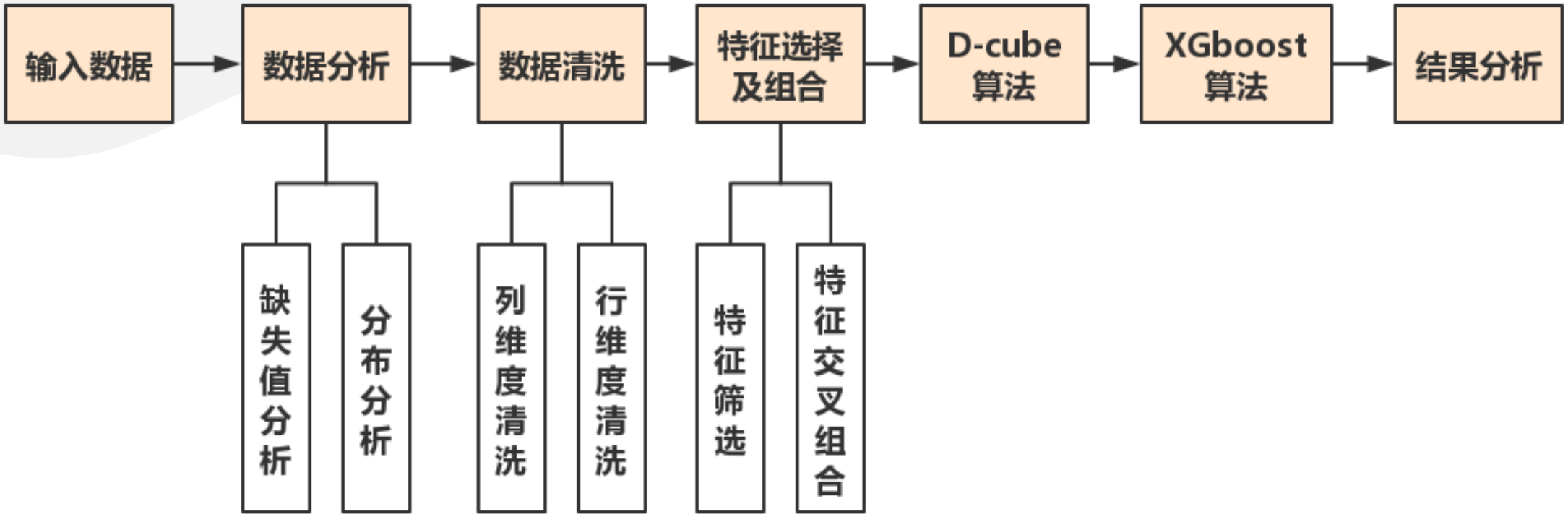
AI技术在反欺诈
和金融风控领域的应用。

项目背景

羊毛党、恶意攻击对手、黑灰产隐藏于众多用户数据之中，给企业造成无法预知的风险隐患。传统黑白名单，规则系统面对变化多端风险数据，力不从心。基于大数据的机器学习方法能够发现数据中的隐藏规律，实现提前预警和主动防御。



项目整体流程图


输入数据
数据分析
缺失值分析
分布分析
数据清洗
列维度清洗
行维度清洗
特征选择及组合
特征筛选
特征交叉组合
D-cube算法
XGboost算法
结果分析

原始数据说明

- 数据量大：

  138w用户

  538w数据

- 多字段种类：

  dtypes: float64(12), int64(1), object(4)

- 多事件类型：

  上传、下载、访问 ...

- 缺失严重

- 各特征重要性不明确

字段	说明
ip	IP 地址
ip_city	IP地址所在城市
email_prefix	邮箱的前缀
email_provider	邮箱的提供商
event_type	事件类型
mobile_prefix_3	手机号前三位
mobile_city	归属城市
time_stamp	事件时间戳
user_name	用户名
user_agent	User Agent
os_version	操作系统版本
resource_owner	资源所属的拥有者
register_type	注册类型
category	生成文章的类别
status	状态
resource_type	资源所属类型
resource_category	资源所属类别

analyse

数据分析

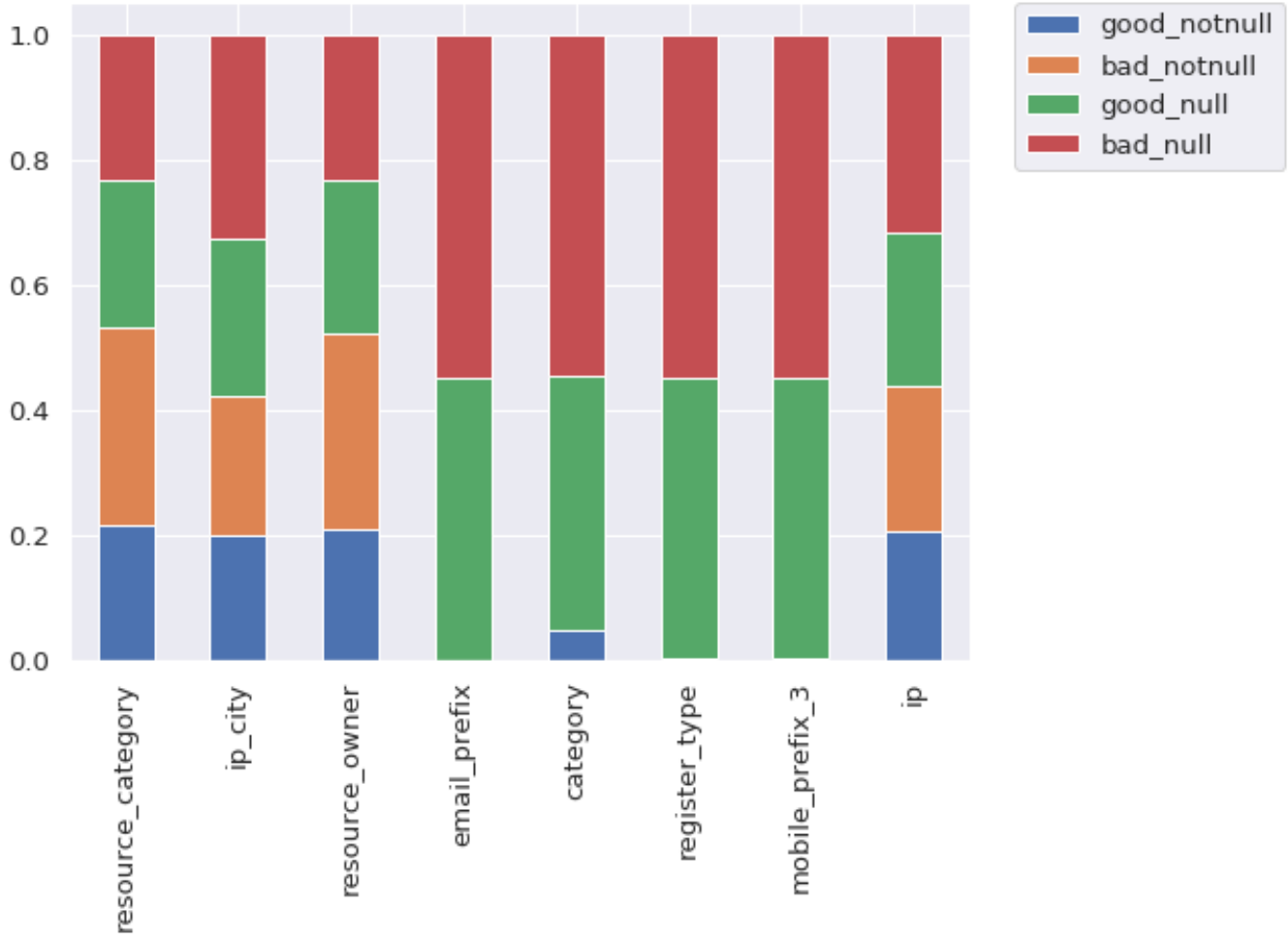
1. 缺失值分析

a.各特征缺失值比例不同

b.好用户/坏用户中缺失值占比存在区别

	Total	Percent
email_prefix	70292	0.999502
email_provider	70292	0.999502
mobile_city	70084	0.996545
mobile_prefix_3	70084	0.996545
register_type	69992	0.995237
status	66785	0.949635
category	66785	0.949635
ip_city	40644	0.577929
ip	39580	0.562799
resource_owner	33562	0.477228
resource_type	32902	0.467843
resource_category	32902	0.467843
event_type	0	0.000000
label	0	0.000000
time_stamp	0	0.000000
user_agent	0	0.000000
os_version	0	0.000000
user_name	0	0.000000

（1）各特征缺失值分布



（2）缺失值在好/坏用户中的分布

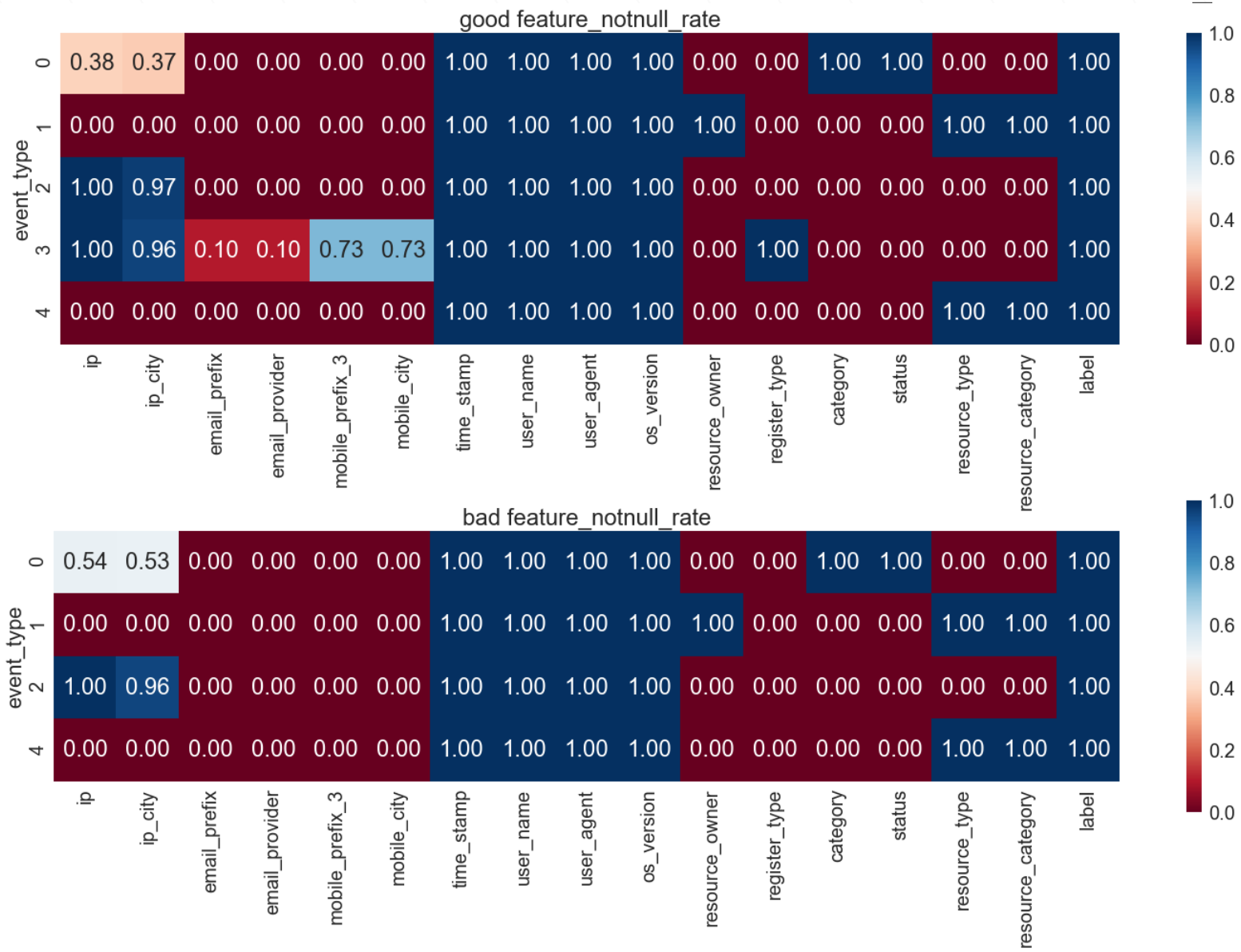
analyse

数据分析

1. 缺失值分析

各事件类型特征缺失情况

（可作为特征筛选的初步依据）



analysis

数据分析

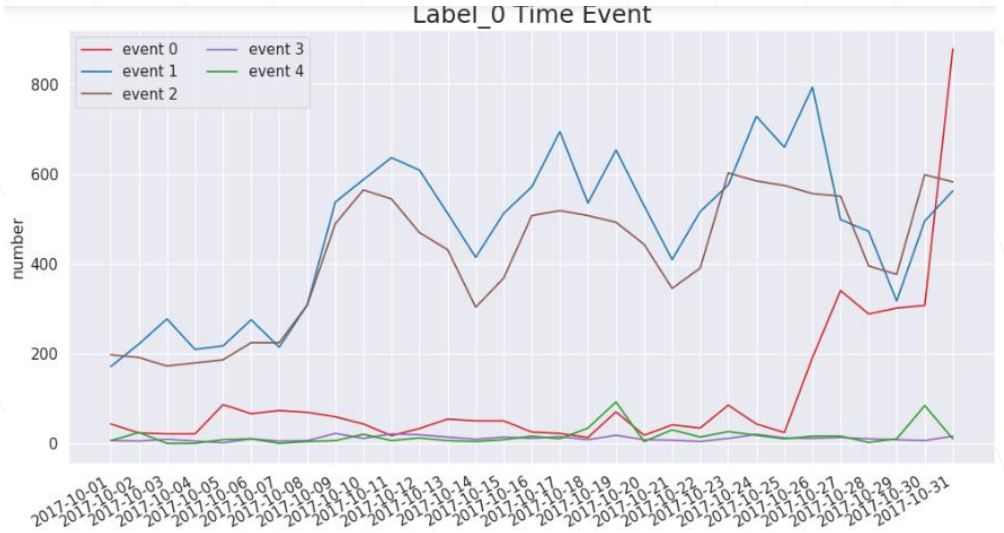
事件分析：

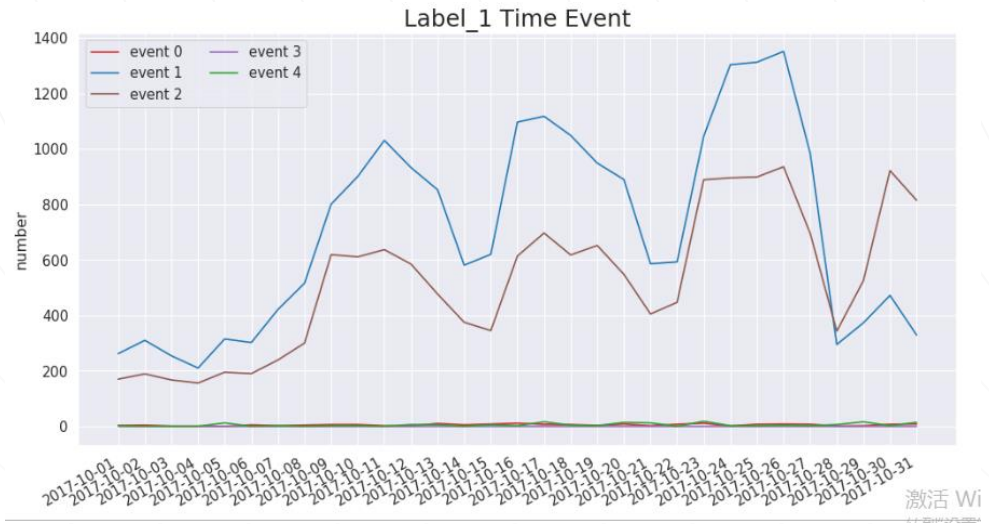
（1）event0在好用户中发生次数较多

（2）event1与event2时间上存在先后关联

（3） event0、event3、event4在整个数据集中占比趋近于0

2. 好、坏账号各事件类型发生次数的时序图





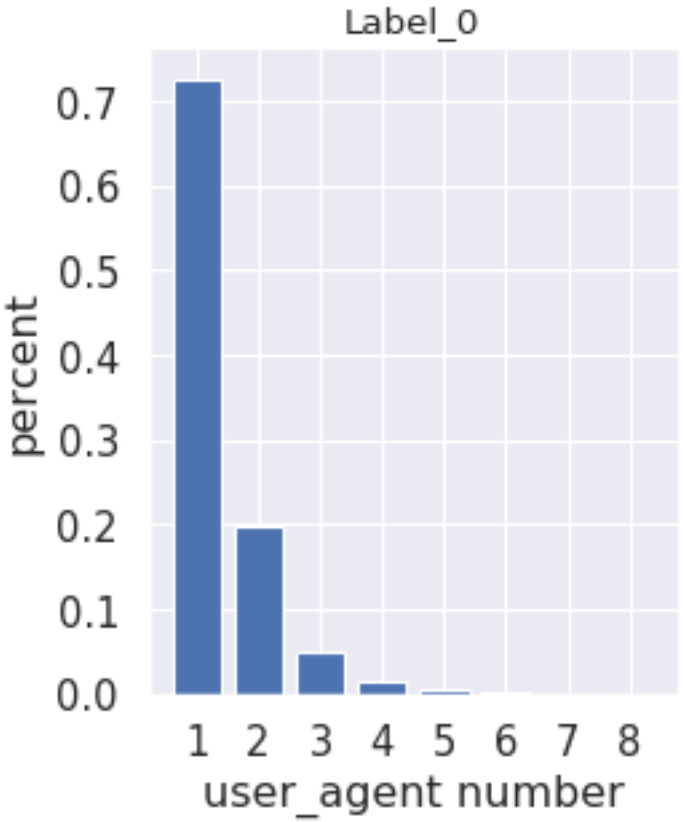
analysis

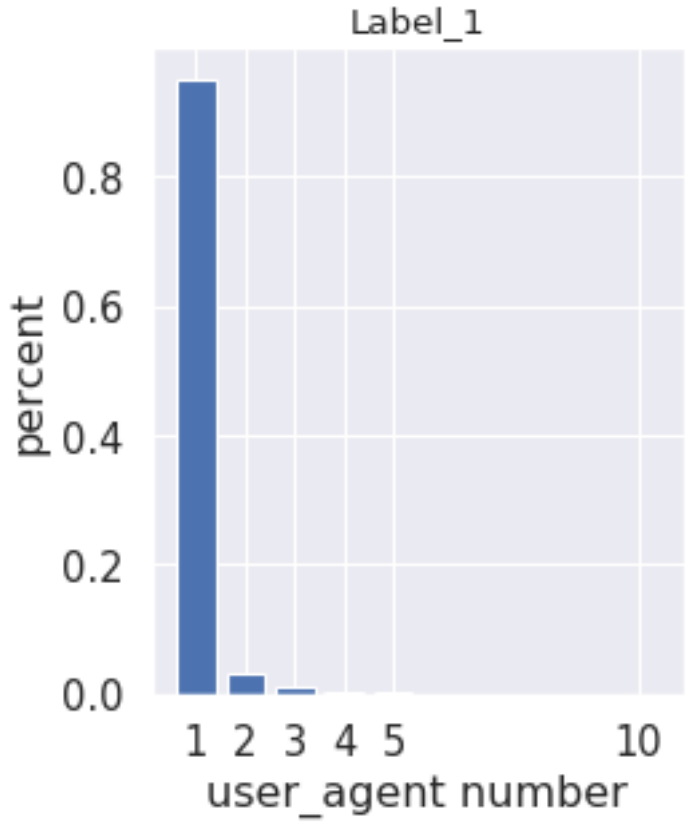
数据分析

正常用户设备使用情况：使用一个user_agent的用户占所有正常用户的73%，使用两个以上代理的用户占比总比19%

异常用户设备使用情况：高达95%的异常用户仅使用一个user_agent，极少的用户处在长尾中

3. 好/坏用户的user_agent使用情况

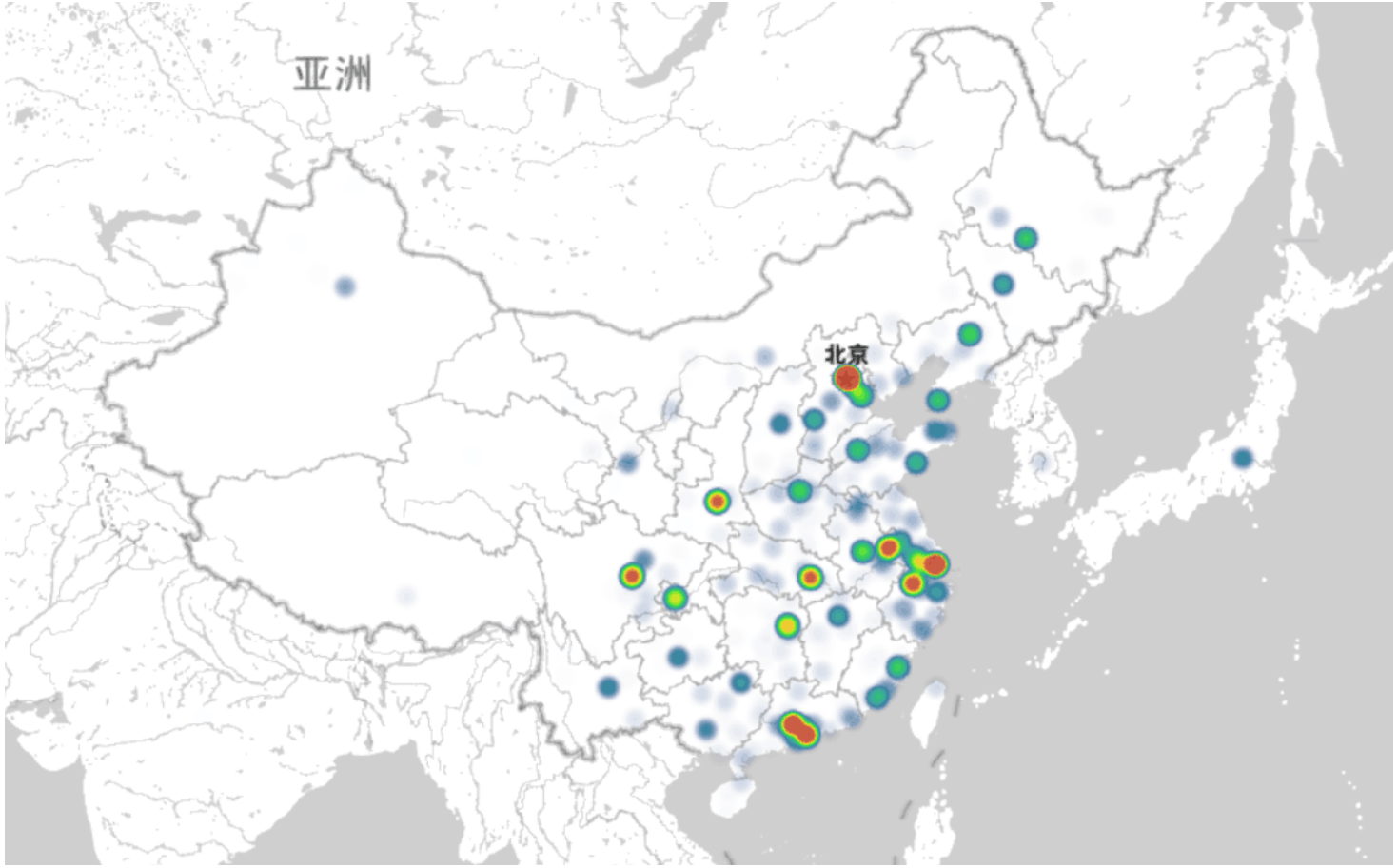




数据分析

4. 用户ip_city分布情况

Ip_city在部分城市体现出集群性



数据清洗

- **1.列维度清洗**

  局域网ip置为空

  ip_city字典编号，全字段ip字典编号

  ip分段、时间分段

- **2.行维度清洗**

  删除event 0,event3,event4事件

  删除重复的行

数据清洗

	ip	ip_1	ip_2	ip_3	ip_4	ip_12	ip_123	ip_1234
70310	106.38.52.170	106.0	38.0	52.0	170.0	106038.0	106038052.0	5355.0
70311	183.12.245.21	183.0	12.0	245.0	21.0	183012.0	183012245.0	3551.0
70312	218.17.250.112	218.0	17.0	250.0	112.0	218017.0	218017250.0	6353.0
70313	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
70314	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

将原数据转换为模型能够读取的类型

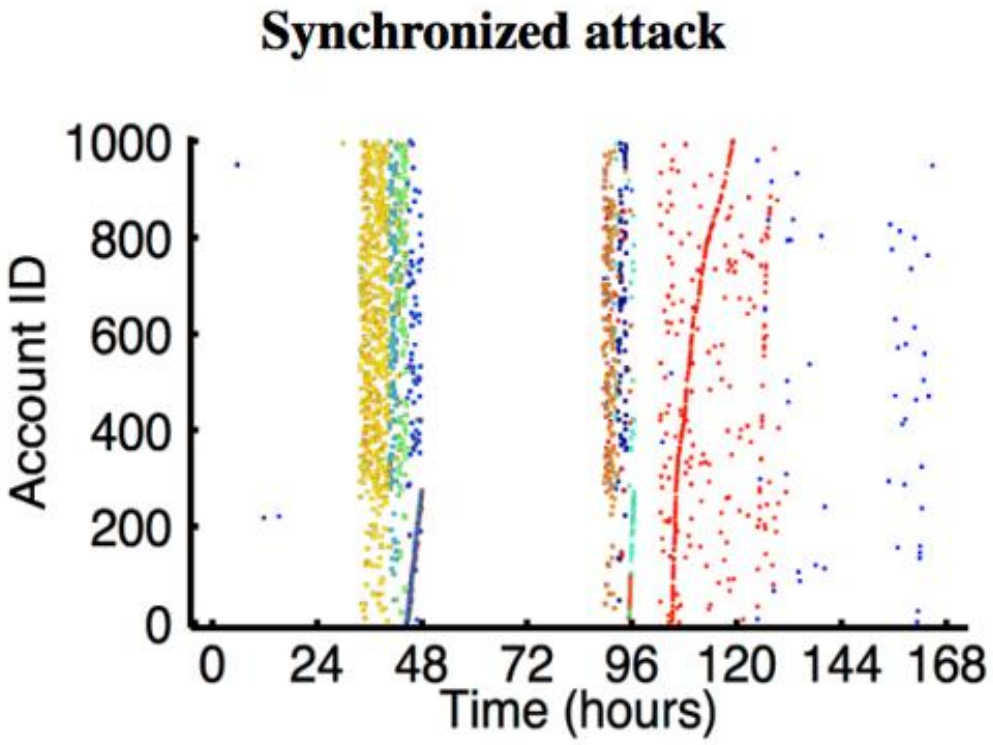
（1）IP分段

	time_stamp	time_stamp_day	time_stamp_hour	time_stamp_3hour	time_stamp_6hour
70310	2017-10-17 10:53:42	17	1710	1709	1706
70311	2017-10-18 22:02:50	18	1822	1821	1818
70312	2017-10-31 15:07:45	31	3115	3115	3112
70313	2017-10-30 19:03:53	30	3019	3018	3018
70314	2017-10-18 01:04:55	18	1801	1800	1800

针对不同时间段对数据进行聚集

（2）时间分段

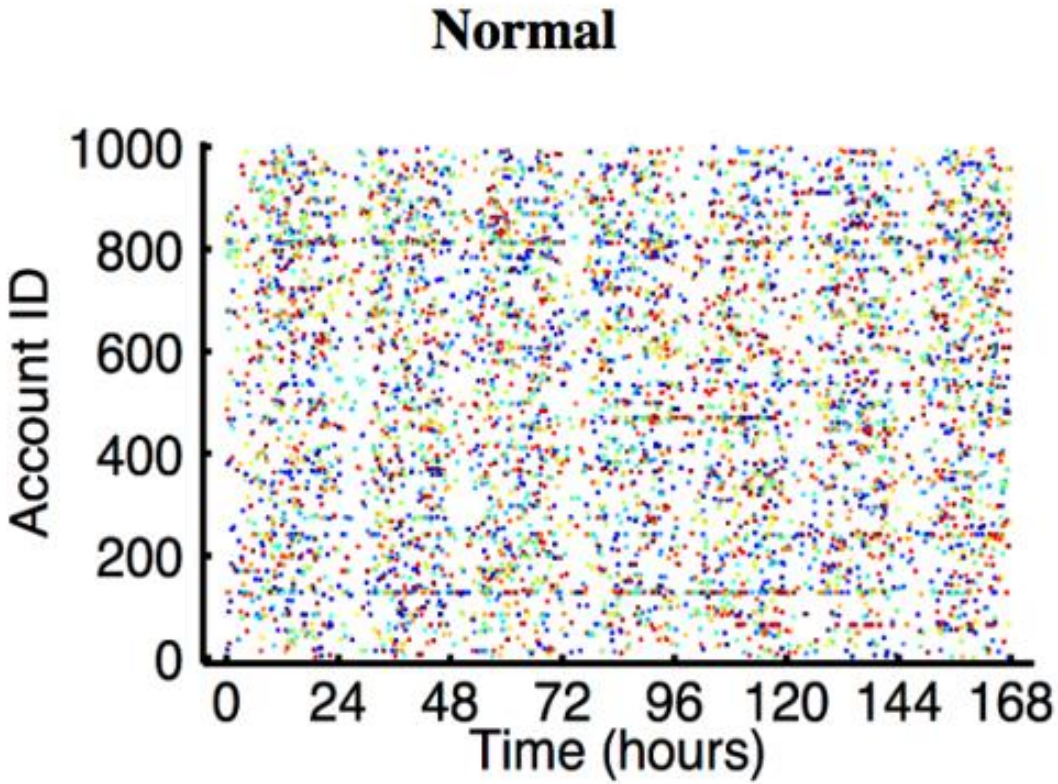
特征工程



（1）异常用户行为时间分布

a) 有大量的无行为时间

b) 行为存在冷启动特征



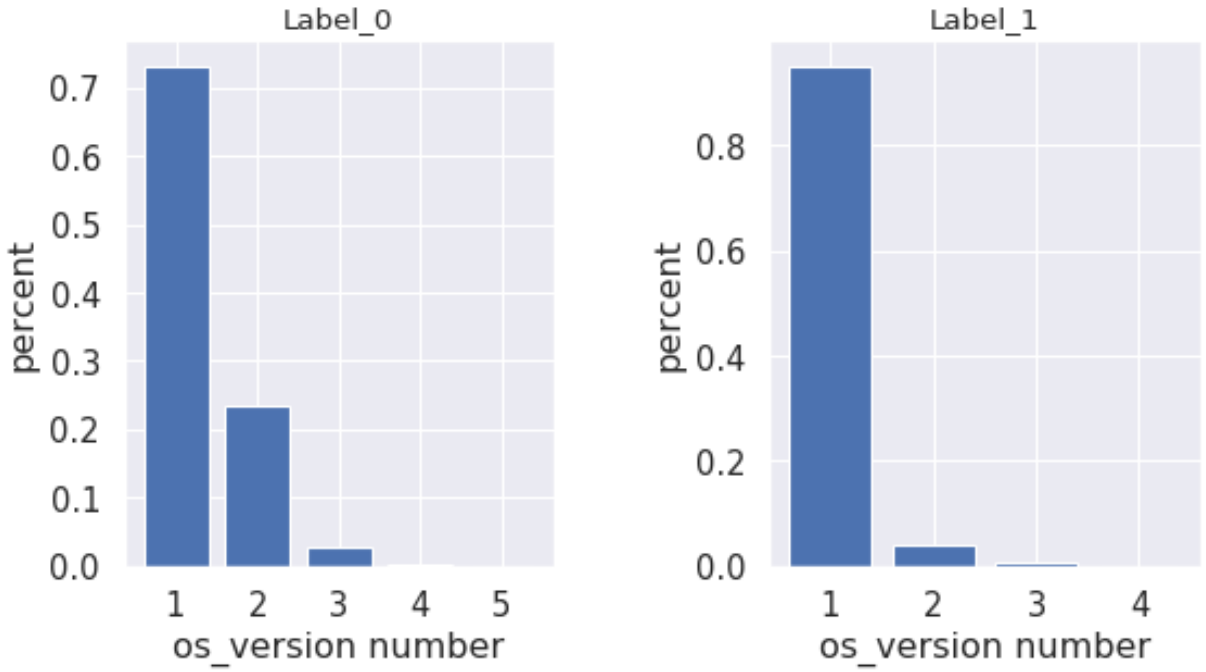
（2）正常用户行为时间分布

a) 行为分散，符合正常情形

b) 无冷启动特征

时间聚集性

特征工程



正常用户使用设备情况： 使用一个os_version的用户占所有正常用户73%，使用两个以上os_version的用户占27%
异常用户使用设备情况： 高达95%的异常用户使用一个os_version，使用两个以上os_version的用户仅占5%

设备聚集性

特征工程

DEECAMP

信息校验

- 对于缺失的基础特征
- …

时间特征

- 当前时间段内，设备出现多少次
- 当前时间段内，该设备出现多少个用户
- 该设备，在当前时间段内出现了多少次行为
- …

IP属性

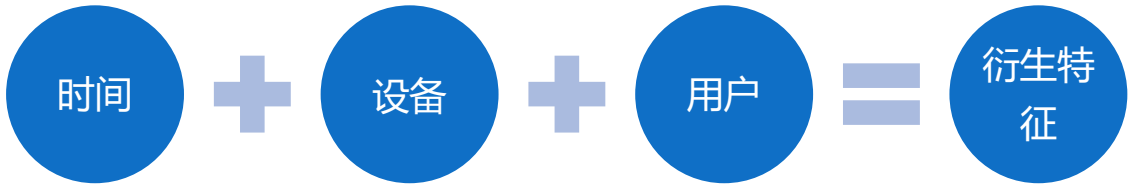
- 当前时间段内，该IP出现了多少个不同的用户
- 当前时间段内，该IP的每个用户出现了多少次
- 该IP被多少不同的device复用。

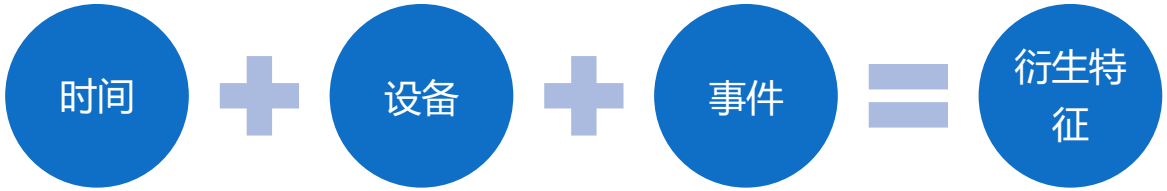
特征组合

DEECAMP

设备字段	时间字段	用户字段	事件字段
● user_agent ● os_version ● ip_1 ● ip_12 ● ip_123 ● ip_1234 ● ip_city ● resource_owner ● resource_type ● resource_category	● time_stamp_day ● time_stamp_hour ● time_stamp_3hour ● time_stamp_6hour	● user_name	● event_type

特征组合





特征组合

DEECAMP

发现可疑用户的特征（共135个）

Feature_combination	Standard	Poly
[user_name, ip_123, time_stamp_6hour, event_type]	index	count
[user_name, ip_1234, time_stamp_day, event_type]	index	count
[user_name, ip_1234, time_stamp_hour, event_type]	index	count
[user_name, ip_1234, time_stamp_3hour, event_t...	index	count
[user_name, ip_1234, time_stamp_6hour, event_t...	index	count
[user_name, ip_city, time_stamp_day, event_type]	index	count
[user_name, ip_city, time_stamp_hour, event_type]	index	count
[user_name, ip_city, time_stamp_3hour, event_t...	index	count
[user_name, ip_city, time_stamp_6hour, event_t...	index	count
[user_name, resource_owner, time_stamp_day, ev...	index	count
[user_name, resource_owner, time_stamp_hour, e...	index	count

发现可疑设备与事件的特征（共175个）

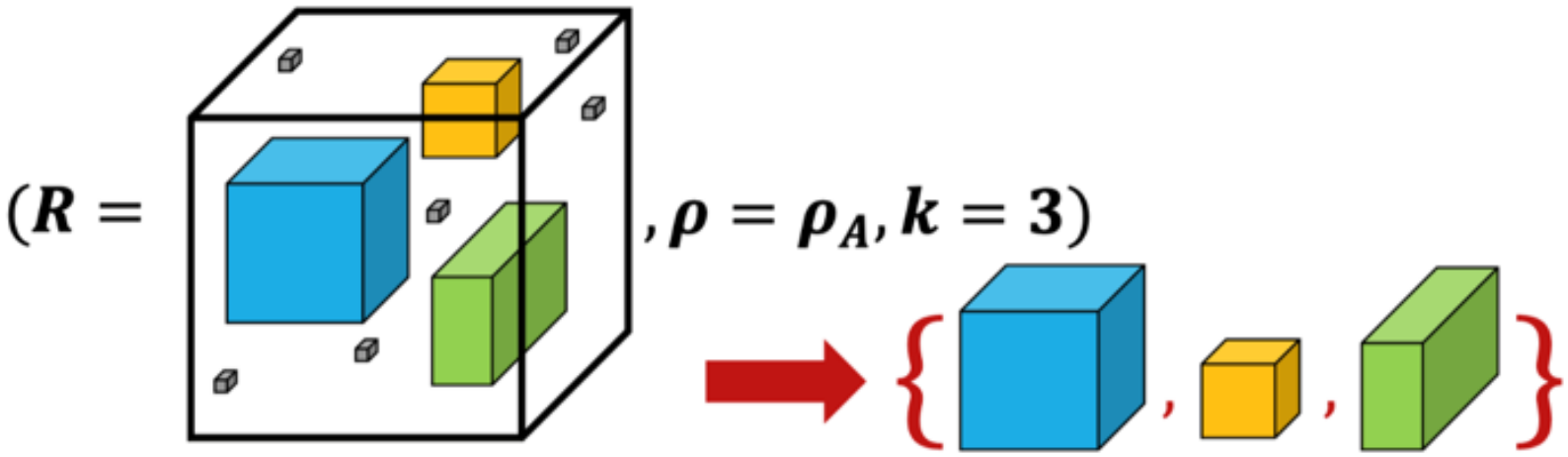
Feature_combination	Standard	Poly
[user_agent]	user_name	nunique
[user_agent, time_stamp_day]	user_name	nunique
[user_agent, time_stamp_hour]	user_name	nunique
[user_agent, time_stamp_3hour]	user_name	nunique
[user_agent, time_stamp_6hour]	user_name	nunique
[os_version]	user_name	nunique
[os_version, time_stamp_day]	user_name	nunique
[os_version, time_stamp_hour]	user_name	nunique
[os_version, time_stamp_3hour]	user_name	nunique
[os_version, time_stamp_6hour]	user_name	nunique
[ip_1]	user_name	nunique

根据所发现的数据分布规则，结合自动化脚本生成交叉特征组合

算法介绍:D-cube

输入　　（1）$\boldsymbol{R}$：输入张量
（2）$\boldsymbol{\rho}$：密度度量
（3）$\boldsymbol{k}$：block的数量

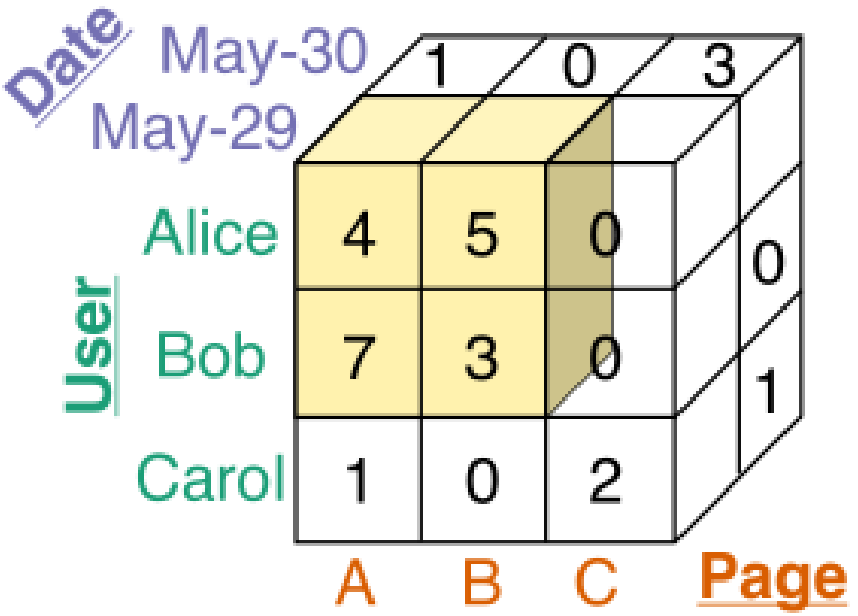
目标：找出使得$\boldsymbol{\rho}$最大的$\boldsymbol{k}$个密度块



算法介绍:D-cube

User	Page	Date	Count
Alice	A	May-29	4
Alice	B	May-29	5
Bob	A	May-29	7
Bob	B	May-29	3
Carol	C	May-30	1
⋮	⋮	⋮	⋮

(a) Relation $\mathcal{R}$



(b) Tensor Representation of $\mathcal{R}$

算法介绍:D-cube

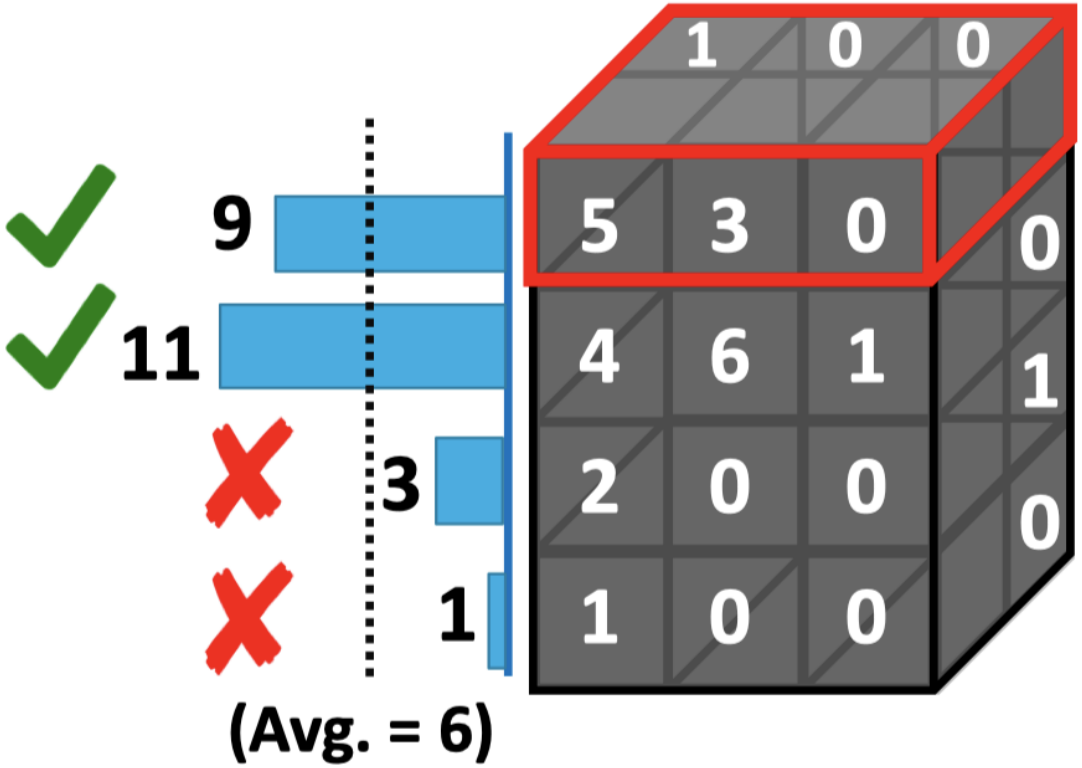
单块检测

步骤1. 从整个张量开始搜索。

步骤2. 选择剩余切片最多的特征维度。

步骤3. 去除质量不超过平均质量的切片。

步骤4. 重复直到张量为空。



算法介绍

XGBoost

- XGBoost是boosting算法的其中一种。Boosting算法的思想是将许多弱分类器集成在一起形成一个强分类器。
- 利用XGBoost可以对输入数据的特征重要性进行排序，从而获得更好的结果。
- 本项目中，运用XGBoost对D-cube的输出结果进行二次学习，从而获得更高的召回率和准确度，同时，实验结果具有可解释性。

结果展示

	Feature id	Precision	Recall
D-cube	42_3	58.87%	33.36%
	40_3	85.10%	0.74%
	64_1	68.26%	17.08%
	149_2	71.28%	20.17%

	Threshold	Precision	Recall
D-cube + XGBoost	0.7	71.0%	45.8%
	0.6	65.9%	70.4%
	0.5	62.4%	87.0%
	0.4	60.2%	90.0%

总结

DEECAMP

1. 查阅资料，理解业务。
2. 认真清洗数据，分析训练集与测试集的分布情况。
3. 机器学习模型构建。

PowerPoint

Thank you

更多内容请访问：

2019

风险罗盘

2019/ 01 / 01