阿里巴巴大数据竞赛

天猫推荐算法大挑战

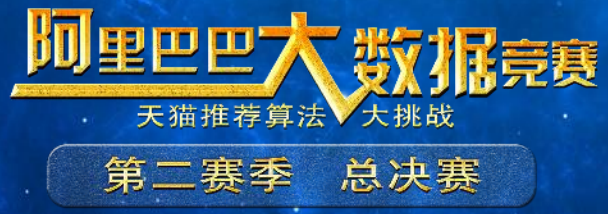
第二赛季　总决赛

Bazinga Team

阿里巴巴大数据竞赛决赛分享

像艺术家一样思考

像建筑工一样实践



我是一名**艺术家**

我沉醉于

天马行空的思考

我热爱

自由自在的创造力...

我是一名**建筑工**

工作中的我严谨规范

一砖一瓦，严丝合缝

方能有

稳如泰山的高楼大厦...

像**艺术家**

一样思考

勇于创新
渴望未知

像**建筑工**

一样实践

脚踏实地
基本功强

优秀的
参赛者 or 研究生
or 算法工程师...

文武双全
敢想敢做

报告提纲

- **团队简介**
- 参赛历程
- 解决方案
- 参赛体会与收获
- 大赛建议

团队简介 – 队名

- **Bazinga** 这个词出自经典三消类游戏泡泡龙中玩家胜利的音效
- 美剧生活大爆炸（The Big Bang Theory）中 Sheldon Cooper 的口头禅，从而流行开来
- 大意为：逗你玩儿，开玩笑的
- 一场说做就做的比赛…
- 对真实大数据场景很感兴趣，参赛心态放松

团队简介 – 成员

张驭宇

中国科学院
计算技术研究所

庞　亮

中国科学院
计算技术研究所

石　磊

中国科学院
软件研究所

团队简介 – 分工

方案设计，框架原型

特征抽取，代码流程化

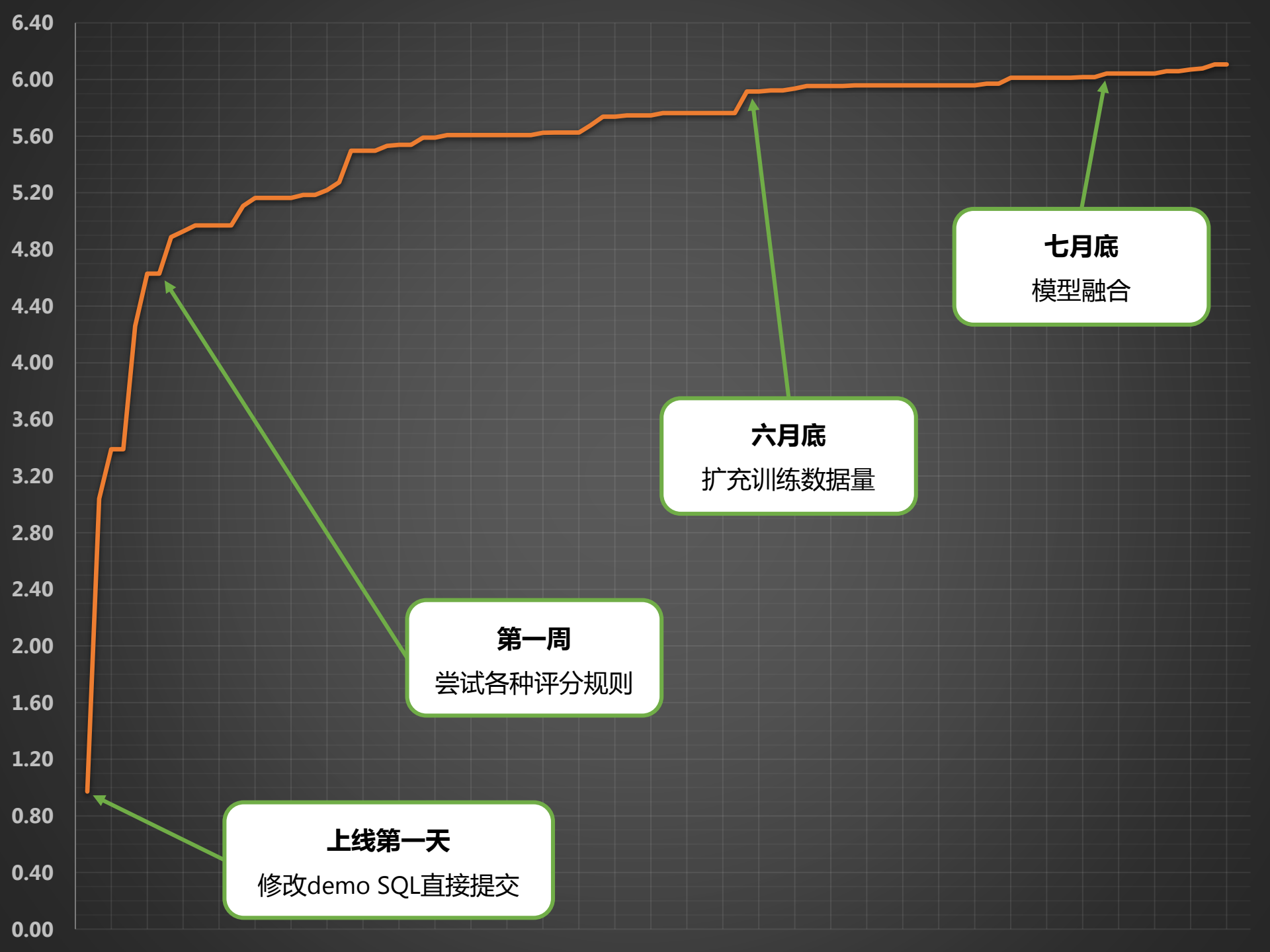
参数调优，模型融合

报告提纲

- 团队简介
- **参赛历程**
- 解决方案
- 参赛体会与收获
- 大赛建议

参赛历程 – 节点

- Season 2 上线第一天
  - 修改demo SQL直接提交
- 第一周
  - 构建本地数据集
  - 尝试各种评分规则
- 六月底
  - 扩充训练数据量
- 七月底
  - 模型融合


6.40
6.00
5.60
5.20
4.80
4.40
4.00
3.60
3.20
2.80
2.40
2.00
1.60
1.20
0.80
0.40
0.00
七月底
模型融合
六月底
扩充训练数据量
第一周
尝试各种评分规则
上线第一天
修改demo SQL直接提交

报告提纲

- 团队简介
- 参赛历程
- **解决方案**
- 参赛体会与收获
- 大赛建议

解决方案 – 比赛任务

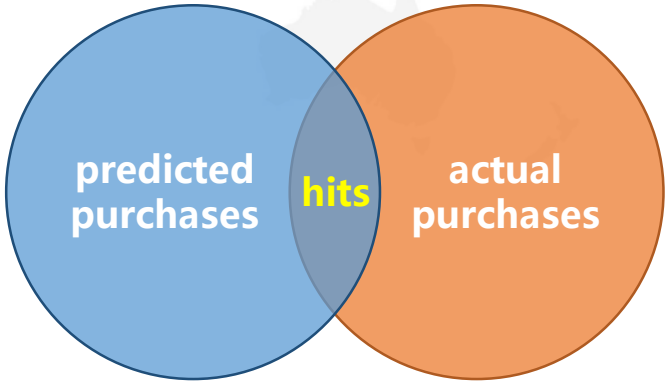
- 数据：天猫用户4个月的行为日志
  - 日志条目格式一致，均为 (user ID, brand ID, action type, date)
- 四种行为类型
  - 点击，购买，收藏，购物车
- 目标问题
  - 预测用户在未来一个月对品牌的购买行为
- 评价指标

$$Precision = \frac{\sum_i^N hitBrand_i}{\sum_i^N pBrand_i}$$

$$Recall = \frac{\sum_i^M hitBrand_i}{\sum_i^M bBrand_i}$$

$$F_1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

即预测哪些 (user ID, brand ID)
在未来一个月会发生购买行为



解决方案 – 数据处理

- 按月划分

Month	From Date	To Date
April	04-15	05-16
May	05-17	06-20
June	06-21	07-18
July	07-19	08-15

- 分月统计数据

	April	May	June	July	Date Total
(user ID, brand ID)	52,909,766	59,299,203	49,776,687	50,188,852	195,303,359
user ID	7,513,602	8,024,680	7,415,736	7,497,824	12,500,984
brand ID	21,305	23,779	25,742	28,036	29,706
actions	138,636,731	159,690,208	139,377,544	134,201,997	571,906,480

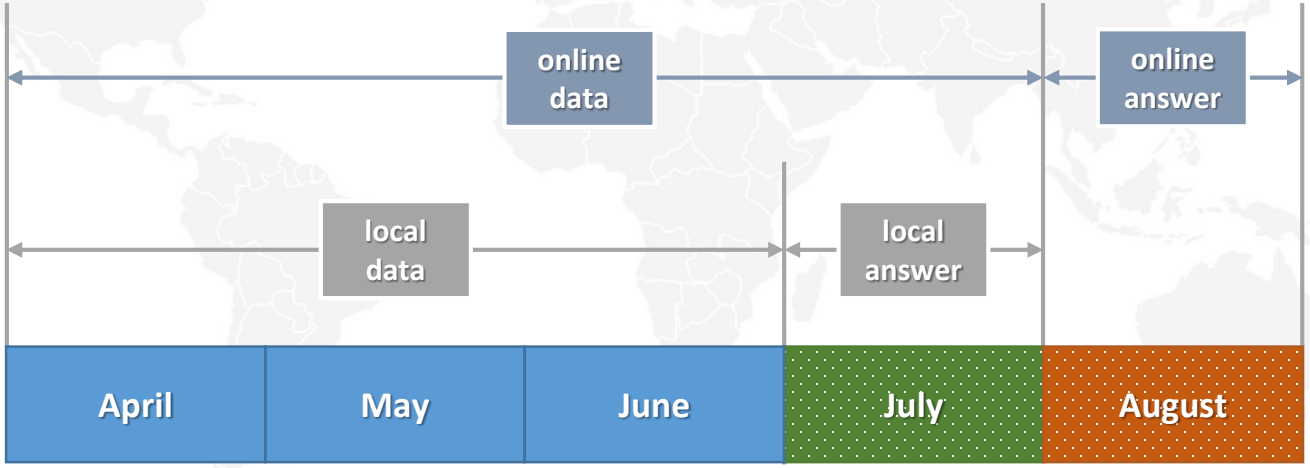
解决方案 – 数据处理

- 数据分析
  - 用户数量千万级，品牌数量万级
  - 4个月数据分布基本均匀
  - 每个用户平均有50次行为，每个品牌平均有20,000次行为
  - 四种行为中最“廉价”的是点击行为
    - 整体点击购买比约为39:1
- 数据清洗
  - 观察到疯狂点击但从不购买的“用户”
  - 使用简单的过滤规则
    - 过滤 $click\# > 500\ and\ buy\# == 0$ 的用户
    - 滤除的数据约占总数据量的4%

品牌层面的信息量相对丰富

解决方案 – 数据处理

- 本地数据集
  - 将七月份的数据作为本地答案集（本地评测前不可见）

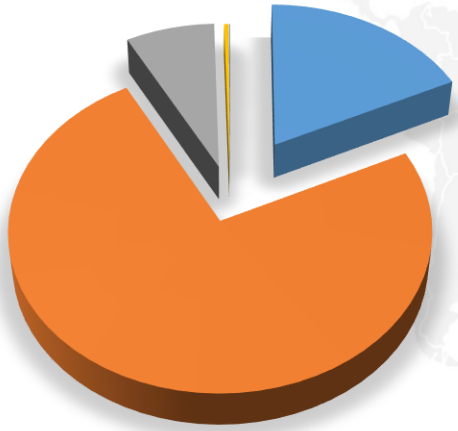


解决方案 – 问题建模

- 条件概率预测
  - 对 (user ID, brand ID) pair 预测

$$P(purchase\ in\ next\ month \mid historcial\ actions)$$

  - $historcial\ actions$ 包含全体用户和品牌的行为



交互pair (17.9%)

非交互pair (74.3%)

user ID, brand ID均来自history

user冷启动的pair (7.4%)

brand冷启动的pair

user和brand均冷启动的pair

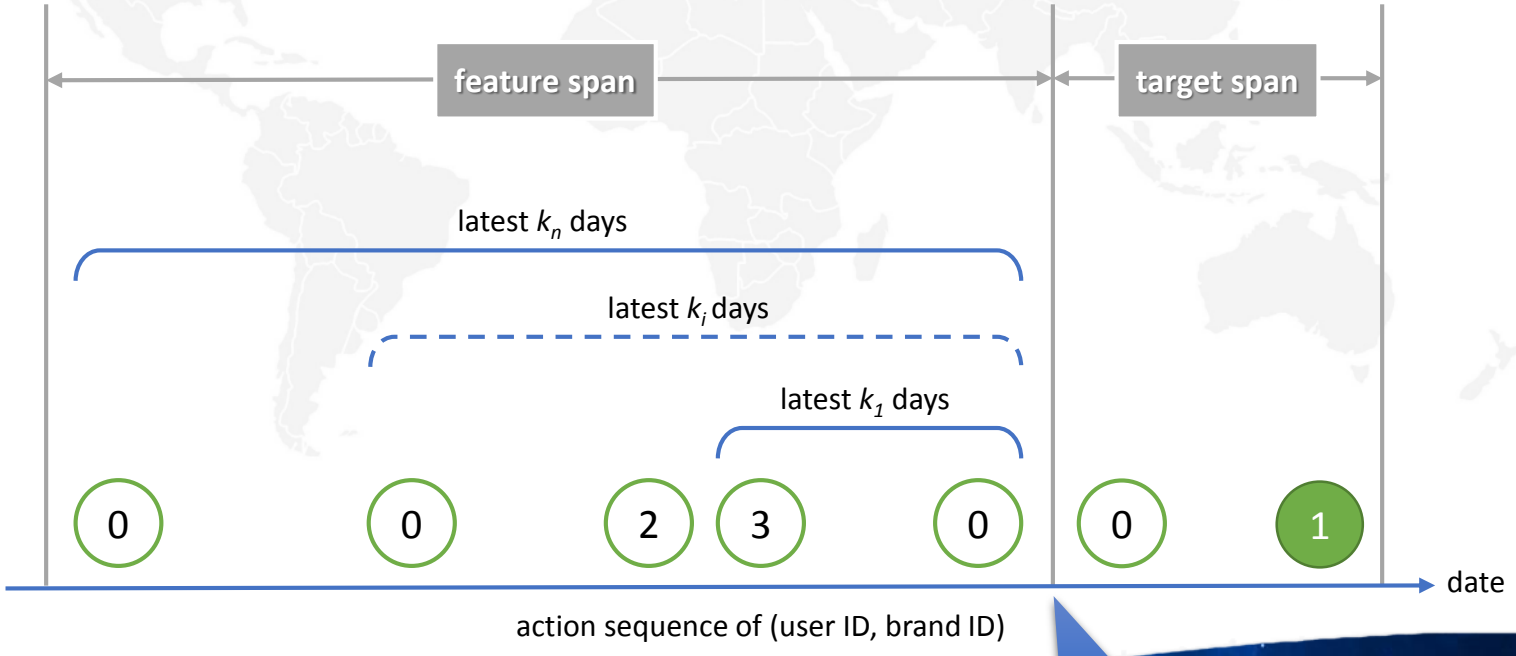
出现新用户或新品牌（冷启动）

在本次比赛中无法预测

local answer集中发生购买行为的pair的构成

解决方案 – 问题建模

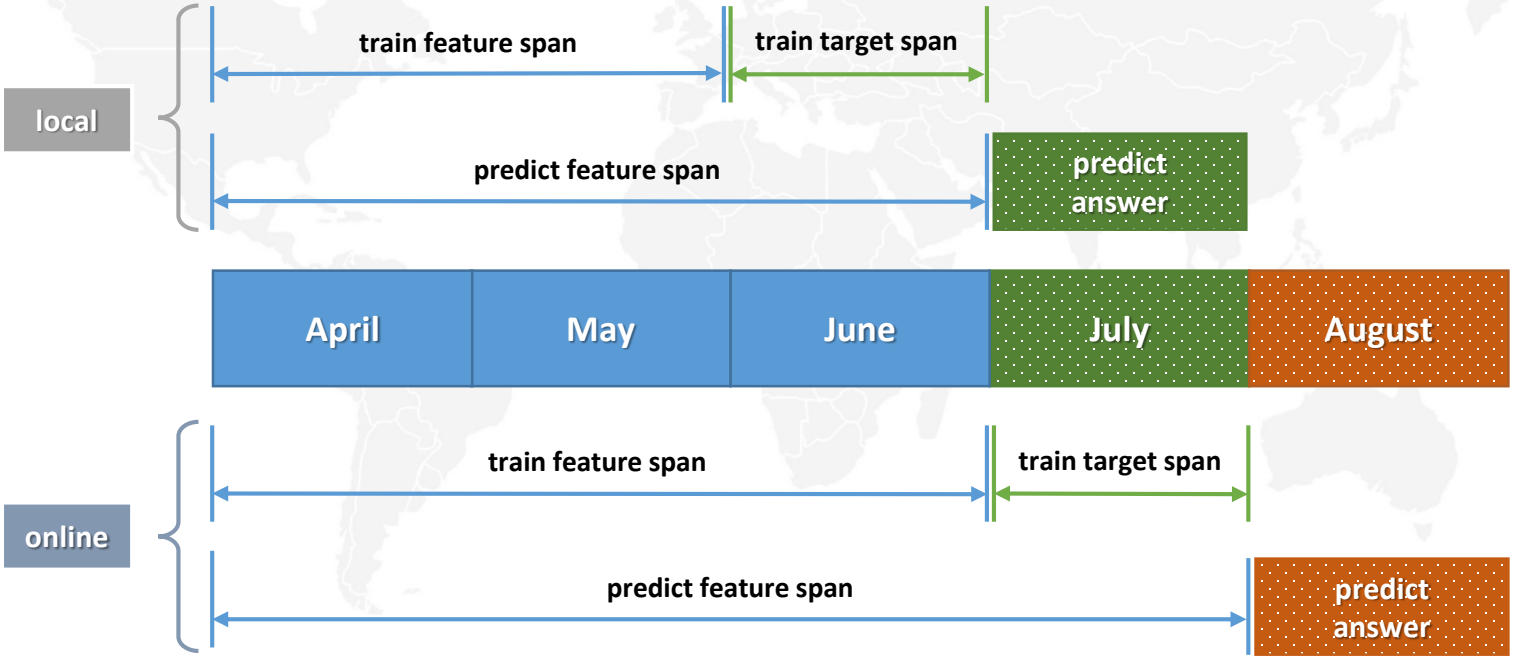
- 样本构造
  - 将 (user ID, brand ID) 的行为序列分割为两部分：
    特征区间 + 目标区间



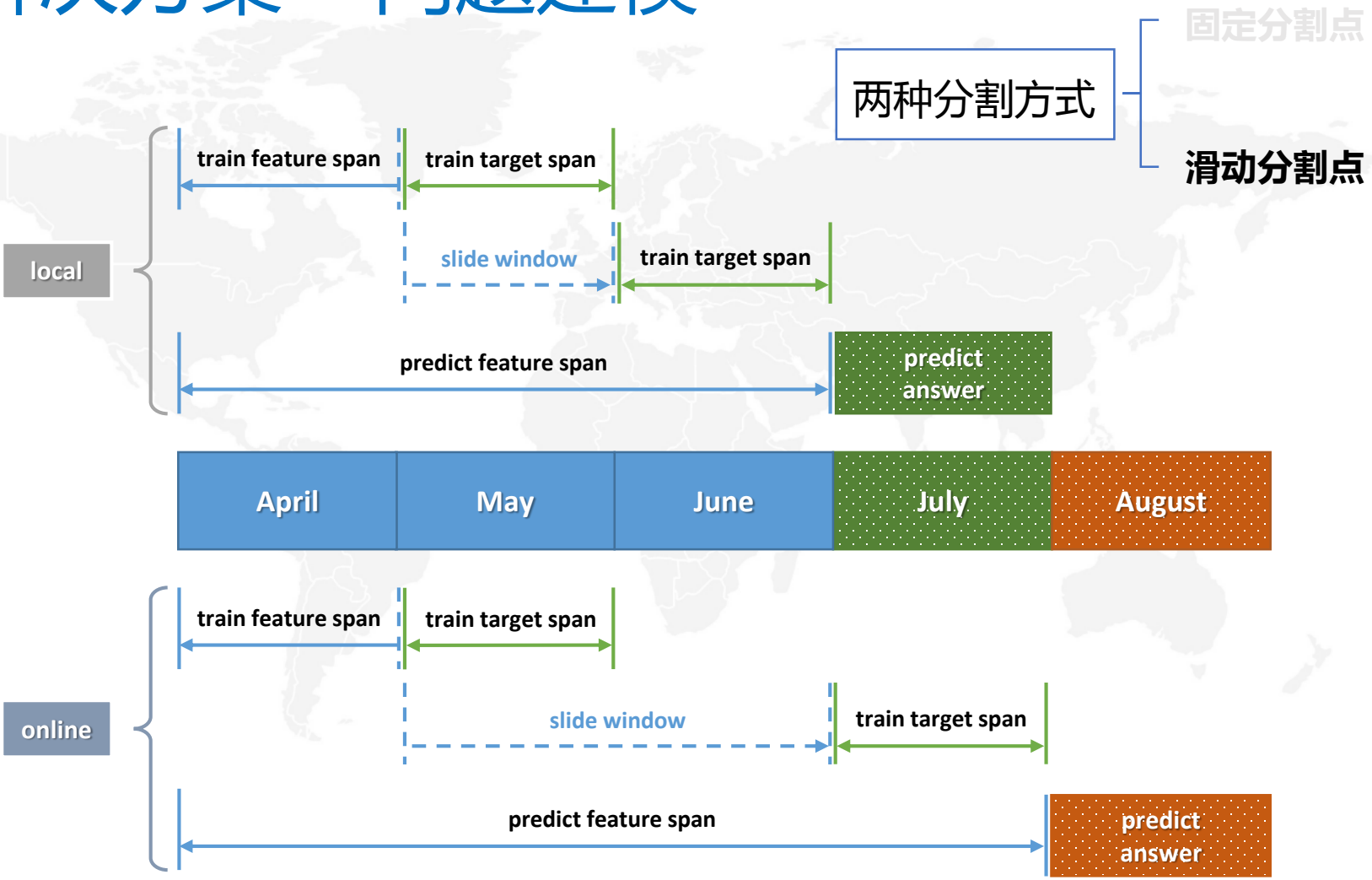
解决方案 – 问题建模

两种分割方式

- 固定分割点
- 滑动分割点



解决方案 – 问题建模


两种分割方式
固定分割点
滑动分割点
local
train feature span
train target span
slide window
train target span
predict feature span
predict answer
April
May
June
July
August
online
train feature span
train target span
slide window
train target span
predict feature span
predict answer

解决方案 – 问题建模

- 采样策略
  - 固定分割点 → 训练样本正负采样比5:25
    - local train正样本约40万，正负样本比约为1:265
  - 滑动分割点 → 训练样本正负采样比1:1
    - local train正样本约1300万，正负样本比约为1:265

模型允许的训练样本规模有限（千万级）

解决方案 – 特征设计

- 主体特征

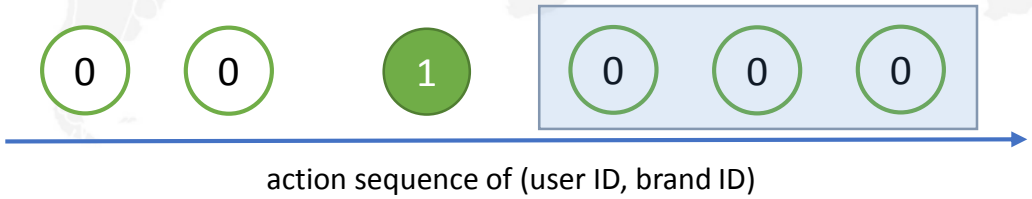
	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#
CVR Features	cvr (buy#, click#)	cvr (buy#, click#)	cvr (buy#, click#)
	cvr (buy day, click day)	cvr (buy day#, click day#)	cvr (buy day#, click day#)
		cvr (distinct buy brand#, distinct click brand#)	cvr (distinct buy user#, distinct click user#)
Ratio Features	ratio (click#, click day#)	ratio (click a#, click b#) X C52	ratio (click a#, click b#) X C52
	ratio (buy#, buy day#)	ratio (buy a#, buy b#) X C52	ratio (buy a#, buy b#) X C52
Flag Features	action flag	action flag	action flag
	other rules		
Global Features	first / last action day	first / last action day	
	first / last buy day	first / last buy day	first / last buy day
	active range length		frequent user ratio

解决方案 – 特征设计

- Counting Features

	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#

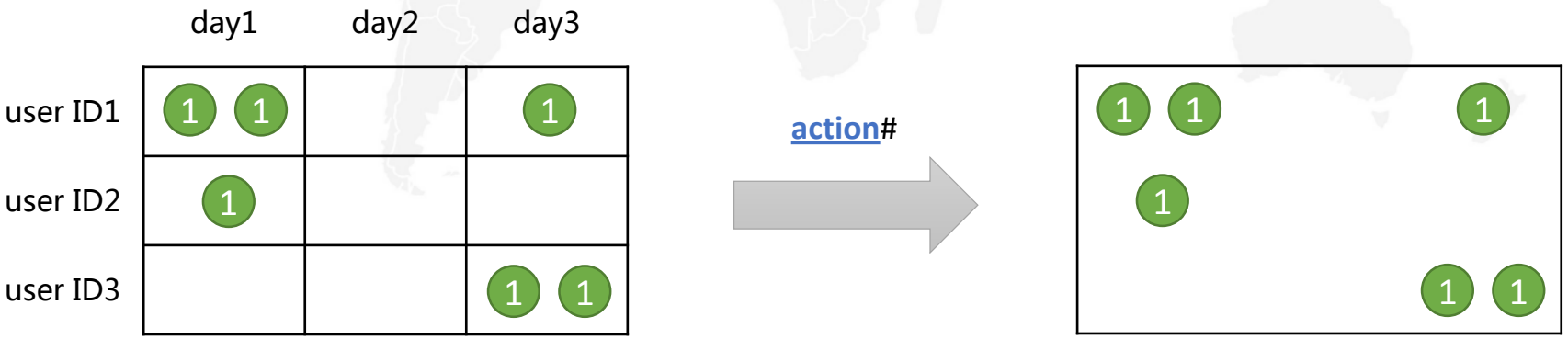
  - valid click



解决方案 – 特征设计

- Counting Features

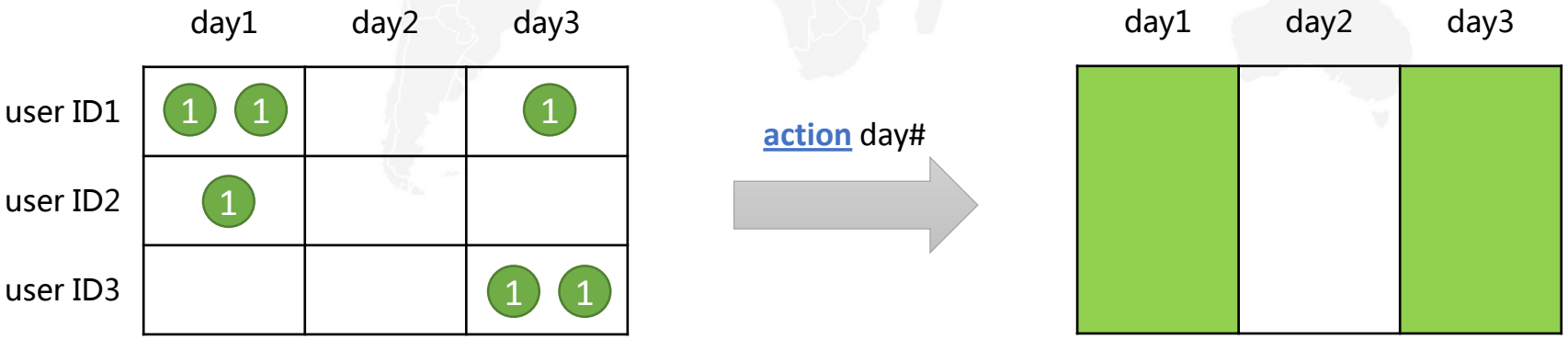
	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#



解决方案 – 特征设计

- Counting Features

	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#



解决方案 – 特征设计

- Counting Features

	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#

解决方案 – 特征设计

- Counting Features

	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#

解决方案 – 特征设计

- Counting Features

	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#

解决方案 – 特征设计

- Ratio Features

	Pair Features	User Features	Brand Features
Counting Features	action#	action#	action#
	action day#	action day#	action day#
	valid click#	distinct action brand#	distinct action user#
	valid click day#	distinct first action brand#	distinct first action user#
		action per brand per day#	action per user per day#
Ratio Features	ratio (click#, click day#)	ratio (click a#, click b#) X C52	ratio (click a#, click b#) X C52
	ratio (buy#, buy day#)	ratio (buy a#, buy b#) X C52	ratio (buy a#, buy b#) X C52

- 上方 brand features 的5种counting，挑选其中2种构造比值特征，共$C_5^2 = 10$ 个
  - ratio (click#, click day#) **该品牌平均每天的点击量**
  - ratio (click#, distinct click user#) **该品牌平均每个用户的点击量**

解决方案 – 特征设计

- Cross Features

	Pair Features	User Features
Counting Features	action#	action#
	action day#	action day#

  - 上方 pair features 和 user features 行内**同种行为**之间相除，构造比值特征
    - ratio (pair click#, user click#) **该品牌的点击量占该用户总点击量的比重**
    - ratio (pair click day#, user click day#) **该品牌的点击天数占该用户总点击天数的比重**
    - …

解决方案 – 特征设计

- Other Features

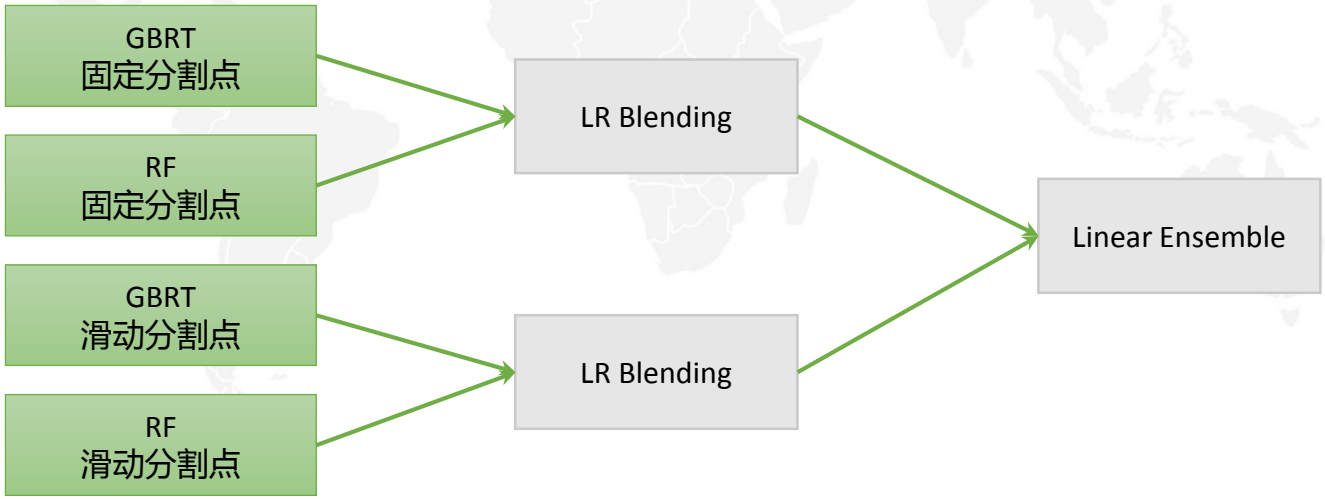
	Pair Features	User Features	Brand Features
CVR Features	cvr (buy#, click#)	cvr (buy#, click#)	cvr (buy#, click#)
	cvr (buy day, click day)	cvr (buy day#, click day#)	cvr (buy day#, click day#)
		cvr (distinct buy brand#, distinct click brand#)	cvr (distinct buy user#, distinct click user#)
Flag Features	action flag	action flag	action flag
	other rules		
Global Features	first / last action day	first / last action day	
	first / last buy day	first / last buy day	first / last buy day
	active range length		frequent user ratio

解决方案 – 模型训练

- 单模型
  - GBRT 固定分割点
  - GBRT 滑动分割点
  - RF 固定分割点
  - RF 滑动分割点

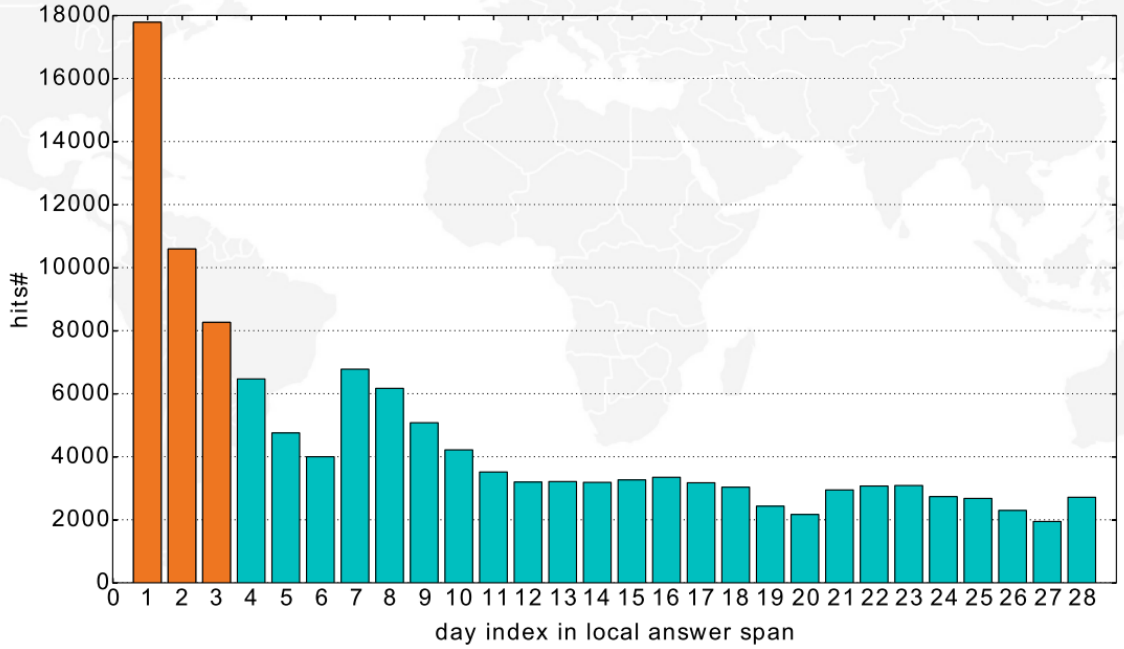
解决方案 – 模型融合

- 两级融合
  - 第一级：单模型级联
  - 第二级：线性融合



解决方案 – 模型能力探究

- 分析local hits的时间分布



解决方案 – 未成年的idea们

- 非交互样本预测
  - 关联 / 序列规则挖掘
  - Top users X Top brands
- 大件商品挖掘
  - 现实中的样本non-i.i.d.
  - 找出大部分用户只买一次的品牌，如果用户已经购买过这样的商品，则不予推荐
- 点击率预测
  - 优点：数据丰富
  - 缺点：噪声较大
  - 在线上系统中有价值

报告提纲

- 团队简介
- 参赛历程
- 解决方案
- **参赛体会与收获**
- 大赛建议

参赛体会与收获 – 团队协作

- 高效的teamwork
  - 团队成员之间共享OneNote笔记本，实时同步
  - 记录比赛中一切需要记录的东东
    - 特征设计表
    - To-do List （进度追踪）
    - 实验结果表
    - …

高效的团队管理

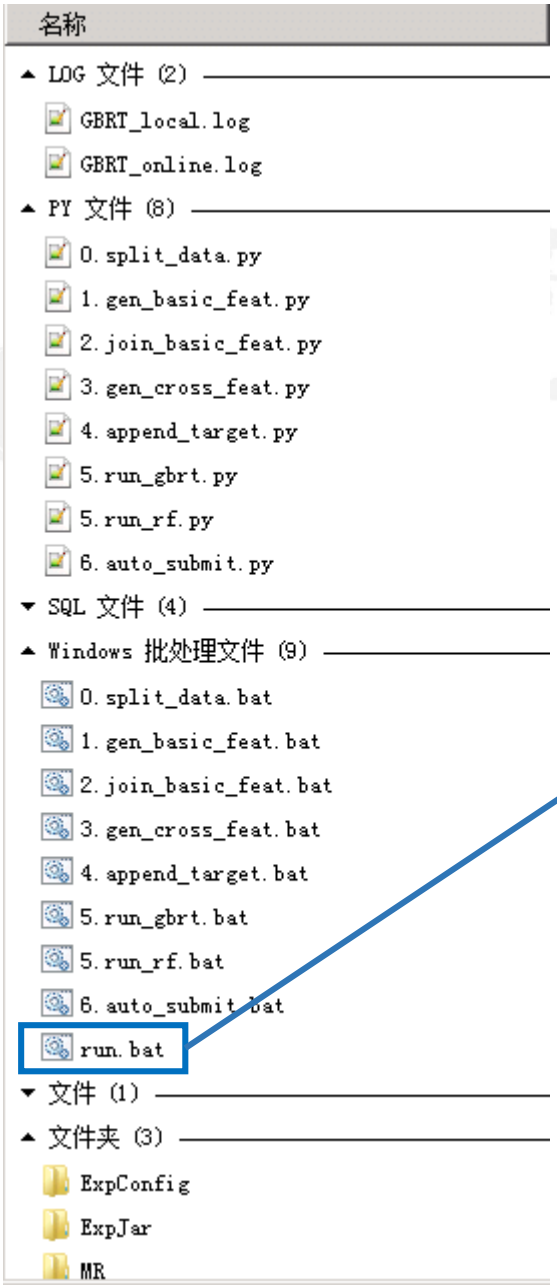
才能 1+1+1>3

参赛体会与收获 – 科学防错

- 流程 foolproof 化
  - 使用Python脚本自动生成SQL脚本
  - 使用bat脚本调用Python脚本

运行总bat脚本，

即可从原始表直接跑到自动提交结果！



run.bat

```
del *.sql

set local_or_online=online
set feat_type=cross
set tag=liang_basic_723_63
set ctag=liang_cross_723

call 0.split_data.bat
call 1.gen_basic_feat.bat
call 2.join_basic_feat.bat
call 3.gen_cross_feat.bat
call 4.append_target.bat

set tree_depth=8
set tree_num=250
set learn_rate=0.08
call 5.run_gbrt_para.bat

set local_or_online=online
set feat_type=cross
set tag=liang_basic_730
set ctag=liang_cross_728

call 1.gen_basic_feat.bat
call 2.join_basic_feat.bat
call 3.gen_cross_feat.bat
call 4.append_target.bat

set tree_depth=8
set tree_num=500
set learn_rate=0.05
call 5.run_gbrt_para.bat
```

参赛体会与收获 – 科学防错

- 流程 foolproof 化
  - 使用Python脚本自动生成SQL脚本
  - 使用bat脚本调用Python脚本
  - 通过config文件向jar包传参（参数用于java中SQL建表以及特征生成）

统一修改参数文件，无需重新导出jar包！

妈妈再也不用担心我的记性~

```
; counting features
counting_pair_feat_cnt  9
counting_user_feat_cnt  8
counting_brand_feat_cnt 18

; ratio features
ratio_pair_feat_cnt 6
ratio_user_feat_cnt 33
ratio_brand_feat_cnt    40

; flag features
flag_pair_feat_cnt  4
flag_user_feat_cnt  4
flag_brand_feat_cnt 4

; Non-bucket features
nonbucket_pair_feat_cnt 6
nonbucket_user_feat_cnt 7
nonbucket_brand_feat_cnt    8

; Time span
time_span   3,7,21,35,63,95

; First label day
first_label_day_train   95
first_label_day_test    123
```

pairDriver.java

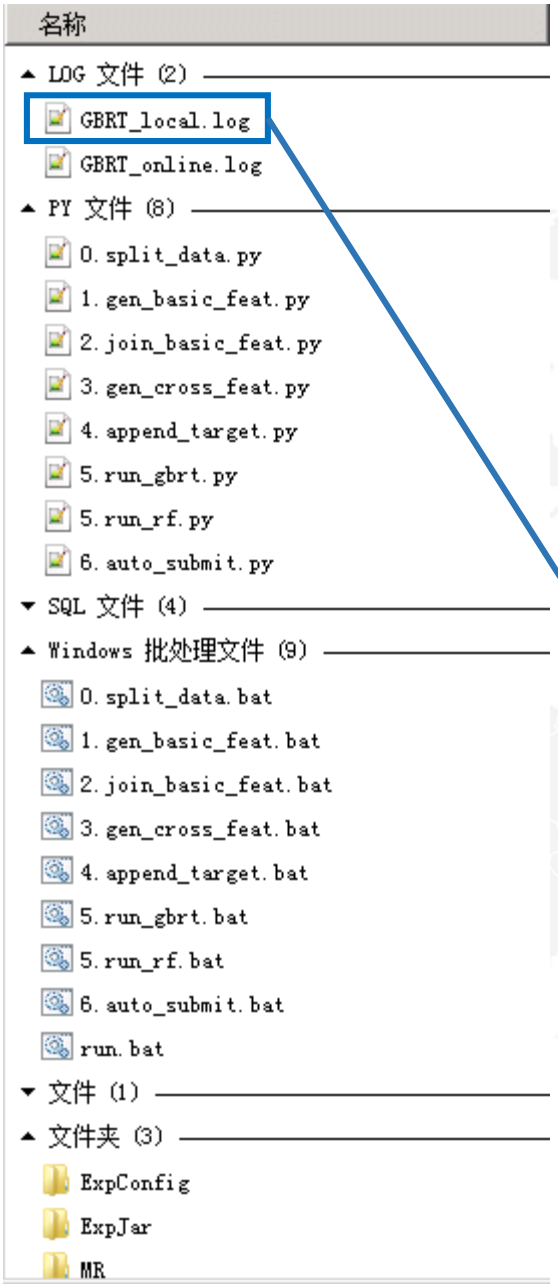
```
20
21  public class pairDriver {
22
23      public static void main(String[] args) throws OdpsException, IOException {
24
25          System.out.println("Generate Pair Features.");
26          System.out.println("Input table: "+args[0]);
27          System.out.println("Output table: "+args[1]);
28          System.out.println("Tag: "+args[2]);
29          System.out.println("First label day: "+args[3]);
30          System.out.println(args[4]);
31
32          Utils.local_or_online = args[4];
33
34          String out_table = args[1];
35
36          JobConf job = new JobConf();
37          job.setLong("first_label_day", Long.parseLong(args[3]));
38          Config.read_params(job, args[2]+".ini");
```

参赛体会与收获 – 科学防错

- 流程 foolproof 化
  - 使用Python脚本自动生成SQL脚本
  - 使用bat脚本调用Python脚本
  - 通过config文件向jar包传参（参数用于java中SQL建表以及特征生成）
  - 自动记录运行log

每一次实验的每一个步骤都有据可查，

根据tag可以找到每一次实验的每一个中间表




```
===============  2014-07-09 14:42:18  ===============
local, tag: liang_flag
feat_type: cross
ptag: liang_flag_liang_baseline
auto_submit: False
Sampling train ... Completed in 343.5s
Total Feature Count: 808
Converting sparse matrix ... Completed in 195.4s
Training ... Completed in 3381.3s
    gbrt_cross_local_model_8_100_10_liang_flag_liang_baseline
Predicting ... Completed in 362.6s
    gbrt_cross_local_predict_8_100_10_liang_flag_liang_baseline
Appending ID ... Completed in 223.8s
    gbrt_cross_local_predict_8_100_10_liang_flag_liang_baseline_append_id
Getting Top-K ... Completed in 181.4s
    gbrt_cross_local_submit_8_100_10_liang_flag_liang_baseline_2300000
Calculating online performance ...
Submit table    gbrt_cross_local_submit_8_100_10_liang_flag_liang_baseline_
Total Pre   2300000
Total Ans   2317304
Total Hit   126168
Precision   5.4856%
Recall  5.4446%
F1Score 5.4650%
```

参赛体会与收获 – 科学防错

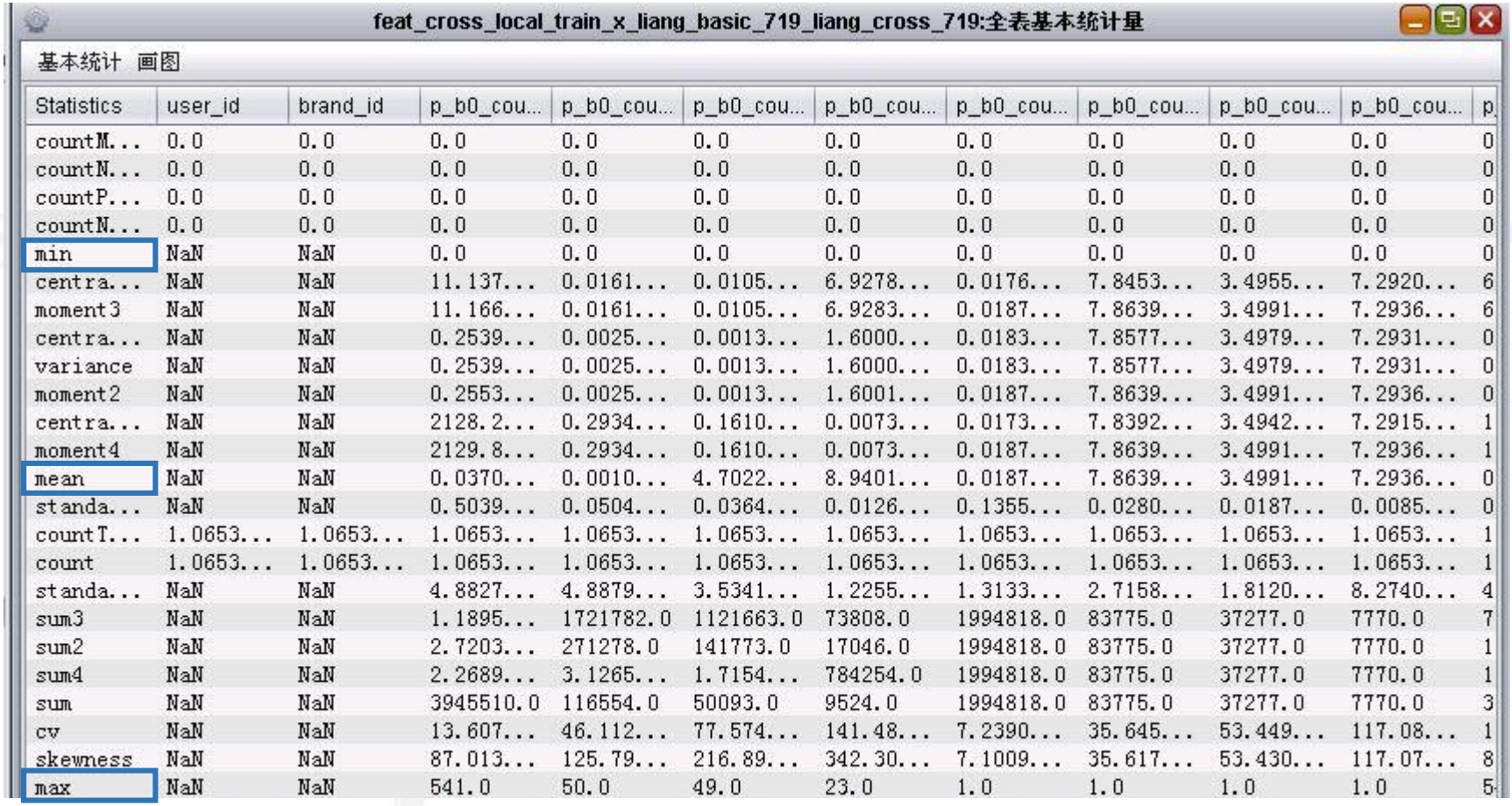
- 流程 foolproof 化
  - 使用Python脚本自动生成SQL脚本
  - 使用bat脚本调用Python脚本
  - 通过config文件向jar包传参（参数用于java中SQL建表以及特征生成）
  - 自动记录运行log

实验流程简明清晰

想出错都难！

参赛体会与收获 – 科学防错

- 多重check，确保bug-free
  - 检查特征值的统计值



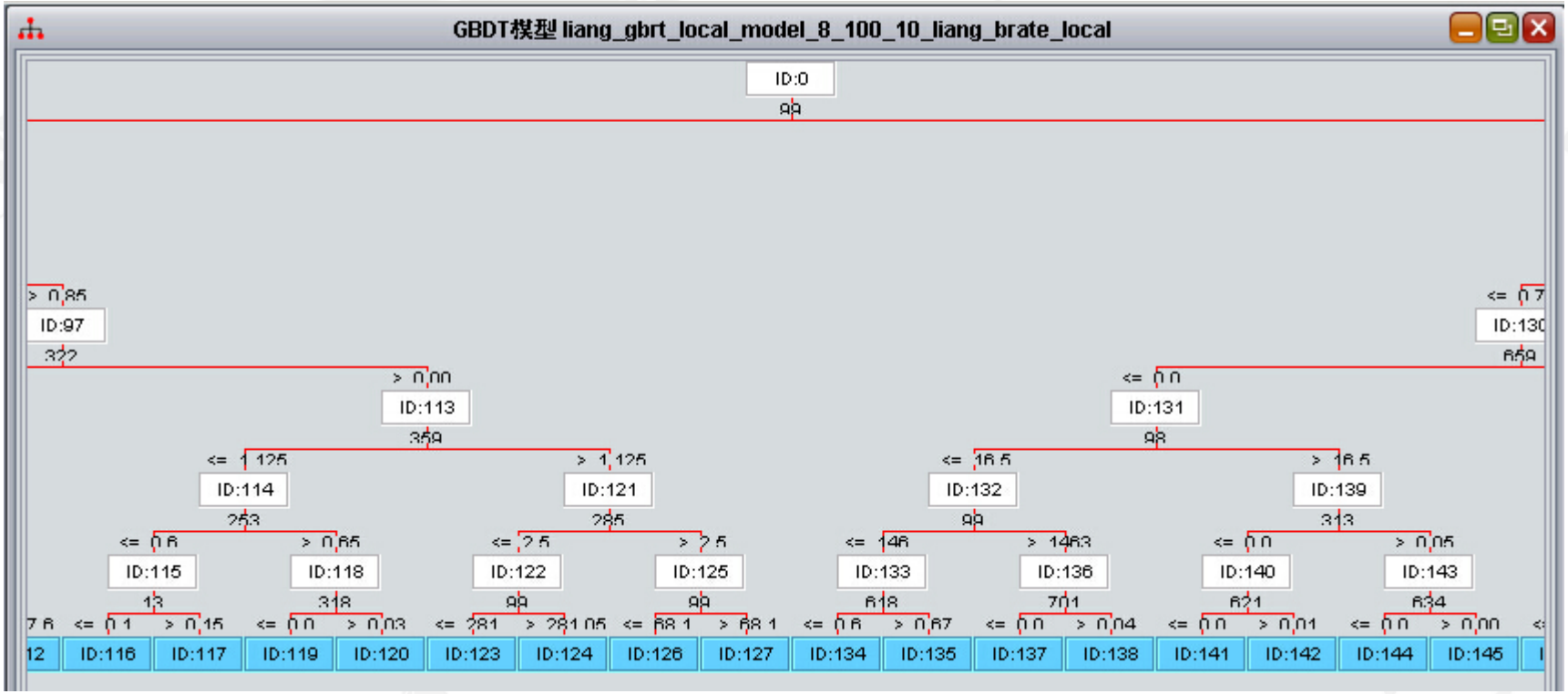
feat_cross_local_train_x_liang_basic_719_liang_cross_719:全表基本统计量

基本统计 画图

Statistics	user_id	brand_id	p_b0_cou...	p_b0_cou...	p_b0_cou...	p_b0_cou...	p_b0_cou...	p_b0_cou...	p_b0_cou...	p_b0_cou...	p
countM...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
countN...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
countP...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
countN...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
min	NaN	NaN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
centra...	NaN	NaN	11.137...	0.0161...	0.0105...	6.9278...	0.0176...	7.8453...	3.4955...	7.2920...	6
moment3	NaN	NaN	11.166...	0.0161...	0.0105...	6.9283...	0.0187...	7.8639...	3.4991...	7.2936...	6
centra...	NaN	NaN	0.2539...	0.0025...	0.0013...	1.6000...	0.0183...	7.8577...	3.4979...	7.2931...	0
variance	NaN	NaN	0.2539...	0.0025...	0.0013...	1.6000...	0.0183...	7.8577...	3.4979...	7.2931...	0
moment2	NaN	NaN	0.2553...	0.0025...	0.0013...	1.6001...	0.0187...	7.8639...	3.4991...	7.2936...	0
centra...	NaN	NaN	2128.2...	0.2934...	0.1610...	0.0073...	0.0173...	7.8392...	3.4942...	7.2915...	1
moment4	NaN	NaN	2129.8...	0.2934...	0.1610...	0.0073...	0.0187...	7.8639...	3.4991...	7.2936...	1
mean	NaN	NaN	0.0370...	0.0010...	4.7022...	8.9401...	0.0187...	7.8639...	3.4991...	7.2936...	0
standa...	NaN	NaN	0.5039...	0.0504...	0.0364...	0.0126...	0.1355...	0.0280...	0.0187...	0.0085...	0
countT...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1
count	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1.0653...	1
standa...	NaN	NaN	4.8827...	4.8879...	3.5341...	1.2255...	1.3133...	2.7158...	1.8120...	8.2740...	4
sum3	NaN	NaN	1.1895...	1721782.0	1121663.0	73808.0	1994818.0	83775.0	37277.0	7770.0	7
sum2	NaN	NaN	2.7203...	271278.0	141773.0	17046.0	1994818.0	83775.0	37277.0	7770.0	1
sum4	NaN	NaN	2.2689...	3.1265...	1.7154...	784254.0	1994818.0	83775.0	37277.0	7770.0	1
sum	NaN	NaN	3945510.0	116554.0	50093.0	9524.0	1994818.0	83775.0	37277.0	7770.0	3
cv	NaN	NaN	13.607...	46.112...	77.574...	141.48...	7.2390...	35.645...	53.449...	117.08...	1
skewness	NaN	NaN	87.013...	125.79...	216.89...	342.30...	7.1009...	35.617...	53.430...	117.07...	8
max	NaN	NaN	541.0	50.0	49.0	23.0	1.0	1.0	1.0	1.0	5

参赛体会与收获 – 科学防错

- 多重check，确保bug-free
  - 检查特征值的统计值
  - 检查特征重要性（在树中的地位）

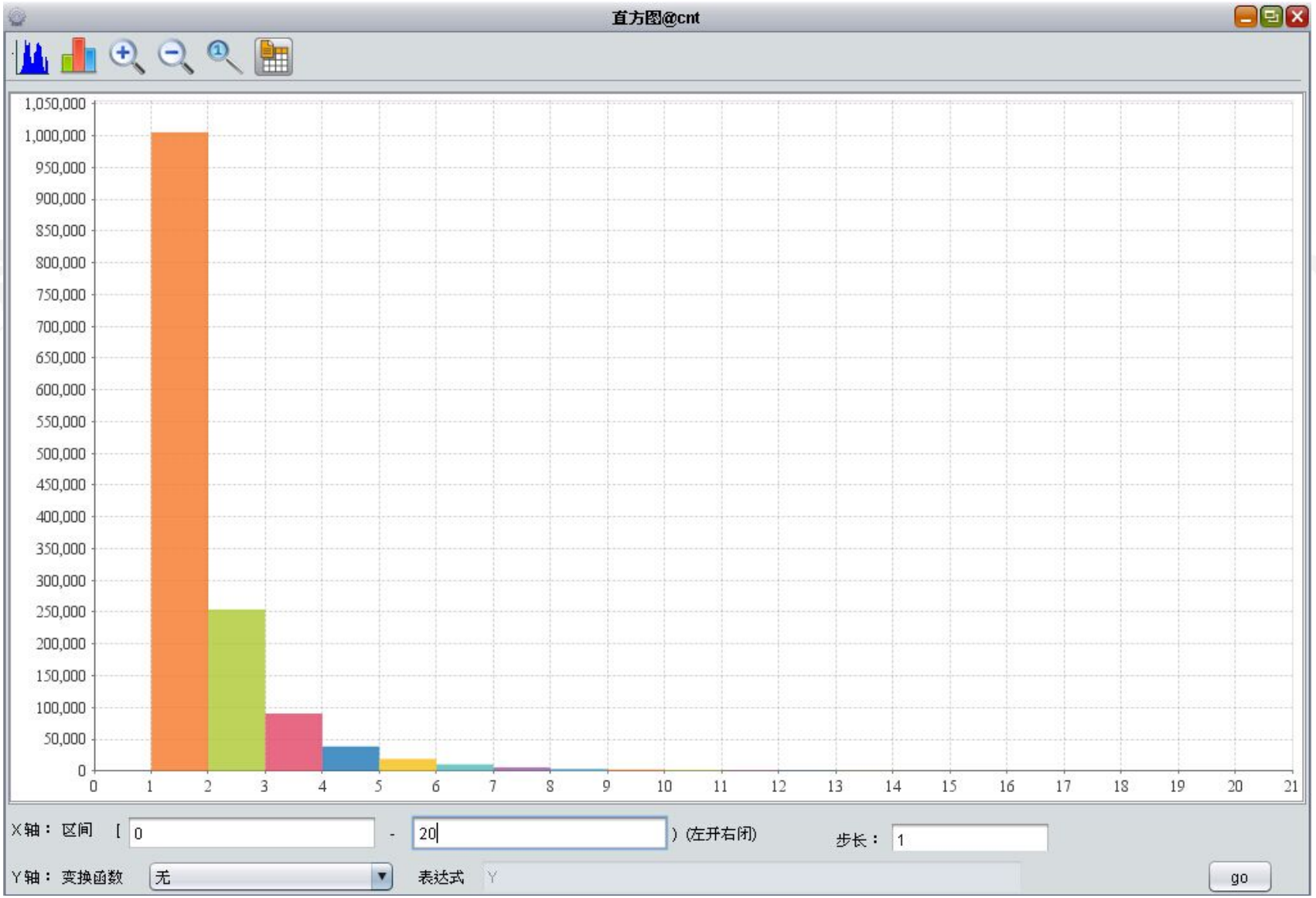


参赛体会与收获 – 科学防错

- 多重check，确保bug-free
  - 检查特征值的统计值
  - 检查特征重要性（在树中的地位）
  - 检查与已有最优提交的重合率

参赛体会与收获 – 科学防错

- 多重check，确保bug-free
  - 检查特征值的统计值
  - 检查特征重要性（在树中的地位）
  - 检查与已有最优提交的重合率
  - 检查各用户的推荐数量分布



参赛体会与收获 – 科学防错

- 多重check，确保bug-free
  - 检查特征值的统计值
  - 检查特征重要性（在树中的地位）
  - 检查与已有最优提交的重合率
  - 检查各用户的推荐数量分布
  - 实验参数和结果及时填表

事无巨细，
建筑工的必备品质

参赛体会与收获 – 与同类竞赛的比较

比赛名称	评价指标	参赛队伍#	获得名次
阿里巴巴大数据竞赛	**F_1Score**	**>7000**	**Top 10**
RecSys Challenge 2013	RMSE	>150	1st place
百度电影推荐系统比赛	RMSE	>300	3rd place
首届中国计算广告学大赛暨RTB算法大赛	Money earned	<100	3rd place
ICDM Contest 2013	NDCG@38	>300	5th place

参赛体会与收获 – 与同类竞赛的比较

- 真实的天猫大数据
  - 对于学生来说，这样的数据很珍贵
- 贴近工业界系统
  - 真实的大数据 + 分布式平台，真刀真枪实战
- 参赛人数创造历史
  - 社会关注度高
  - QQ讨论群近1200人，官方旺旺群近600人
- 多轮比赛 + 末位淘汰制
  - 比赛周期略长

参赛体会与收获 – 对竞赛本身的思考

- Exploitation >> Exploration
  - 过于侧重retargeting
  - 假设线上系统使用real-time counting…
  - 精度确实会大幅提升，意义也许没有看起来的提升那么大
    - 例如，利用用户几秒钟前的点击预测购买，用户此时并不依赖推荐
- 真实系统应该帮助用户发现新的兴趣点
  - 非交互
  - 冷启动
    - 全新的用户
    - 全新的品牌 / 商品

报告提纲

- 团队简介
- 参赛历程
- 解决方案
- 参赛体会与收获
- **大赛建议**

大赛建议

- 给定budget限制，用完即止
  - 优雅地将效率纳入评价体系
  - 环保节能，更贴近实际系统
- 提供更丰富的数据
- MR job的可视化需要加强
- 比赛周期可适当缩短
- 奖励力度和范围可扩大

致谢

- 感谢阿里巴巴举办这次比赛，提供数据和平台
- 感谢所有为比赛辛苦忙碌的组织者们，尤其是天渡哥、山水哥和一婷姐，你们辛苦了！请求给这几位升职加薪！
- 感谢共同奋战过的所有参赛选手

Thanks!

新浪微博: **@张驭宇UCAS**

Email: i@zhangyuyu.com

MR job的可视化畅想