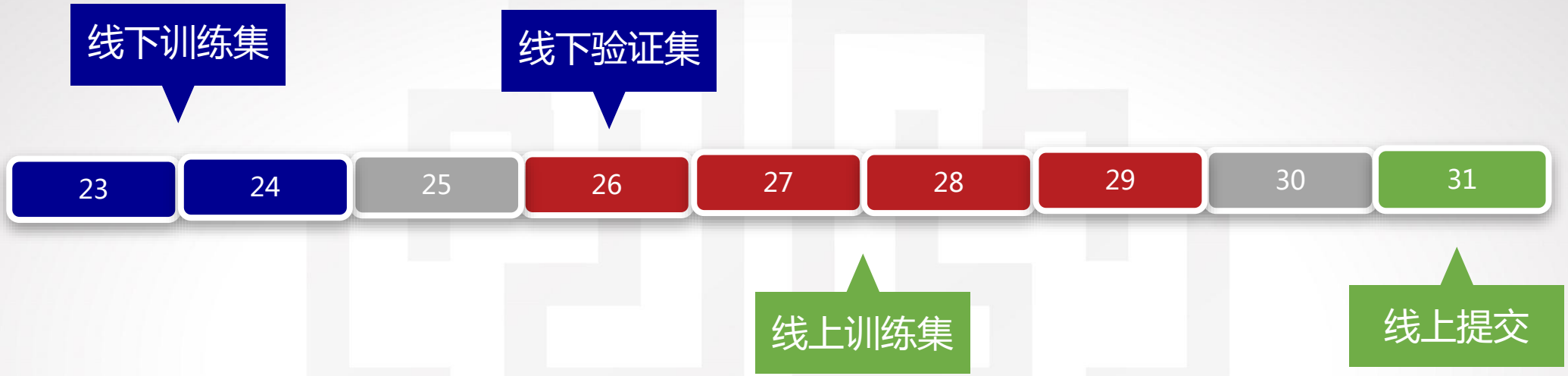
腾讯社交广告
Tencent Social Ads

高校算法大赛

01 团队介绍

02 特征工程

样本划分/Sample Split



- 划分样本要保证线上线下同增同减
- 由于30号中有较大比例的样本Label是不准确的，线上训练集舍弃；同时线下使用26号做验证时也要从训练集中舍弃25号的样本
- 线下训练集训练时需要将convertionTime大于26号的label全部置为0

基础特征/Base features

点击当日特征/Today features

非LEAK特征

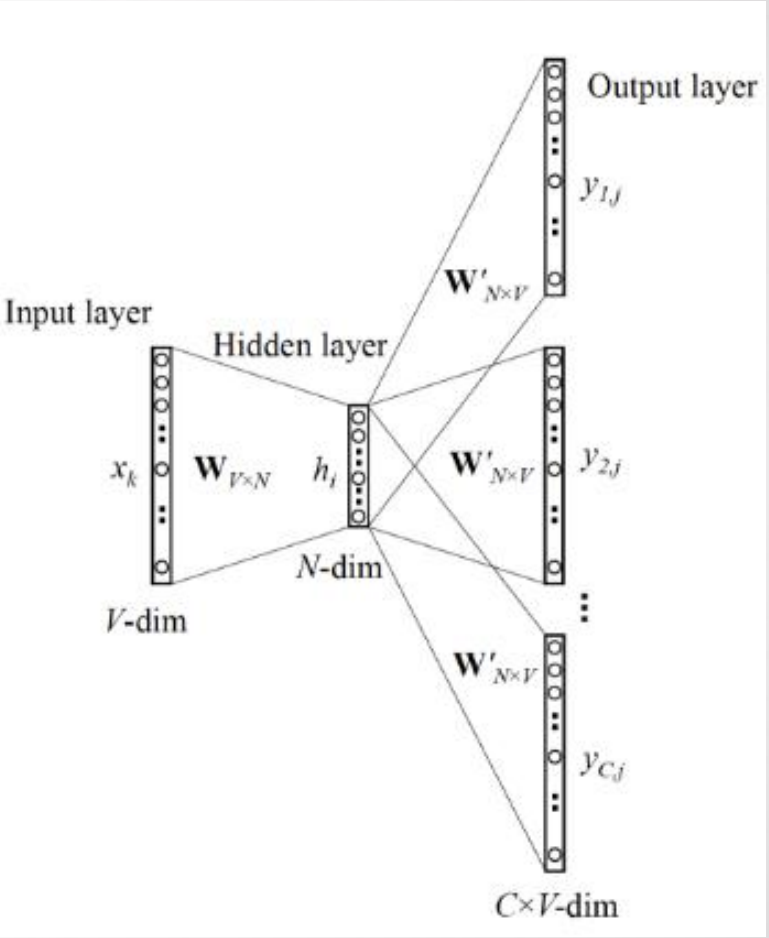
- 当前是否是用户今天第一次点击
- 当前是用户今天第几次点击该APP/CreativeID/PositionID
- 用户在此前1个小时内点击的统计量
- 此前1小时内各个ID的点击量

LEAK特征

- 当前是否是用户今天最后一次点击
- 今天内各个ID的点击量
- 今天内2类ID组合的点击量
- 距离用户下一次点击APP/CreativeID/PosiitonID的秒数

Word2Vec

- 使用user_app_action表
- 前提假设：一起安装的app具有相似性
- 生成APP的词向量（10维）
  - 利用用户安装历史记录，统计每个用户安装app列表
  - 每个用户的安装app按照时间顺序排序
  - 每个用户的安装app当做一篇文章，每个appid 作为一个词，利用滑动窗口扫描每篇文章，训练两层的神经网络，每个窗口的中心位置的词去预测上下文的词，损失函数为负的最大似然。取第一层的参数作为词向量。
- 直接将词向量加入LightGBM的特征进行训练



离散类特征/Dispersed features

转自连续型特征

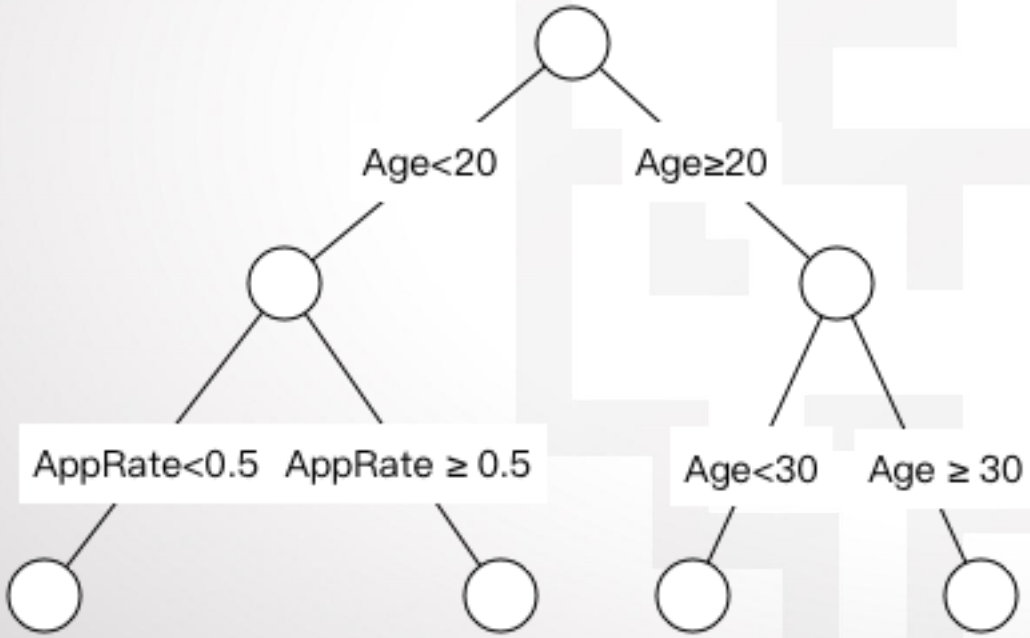
- 转化率
- 组合转化率
- 点击数、转化数
- 用户特征
- APP安装（转化）的统计特征
- 当日统计特征
- FFM、LR等模型使用连续型特征前需要对特征进行分箱

离散型特征

- UserID 、 AppID 、 CreativeID 、PositionID等
- 用户历史点击过的APP
- 用户历史点击过的APPCategory
- 将出现次数较少的ID直接归为一类

连续型特征分箱/Bin features

XGBOOST训练出来的决策树在每个树节点分裂时选择的是最优的分裂点，这些分裂点是经过树模型选择出来的最优结果，我们可以直接借助这些分裂点对连续特征分箱。

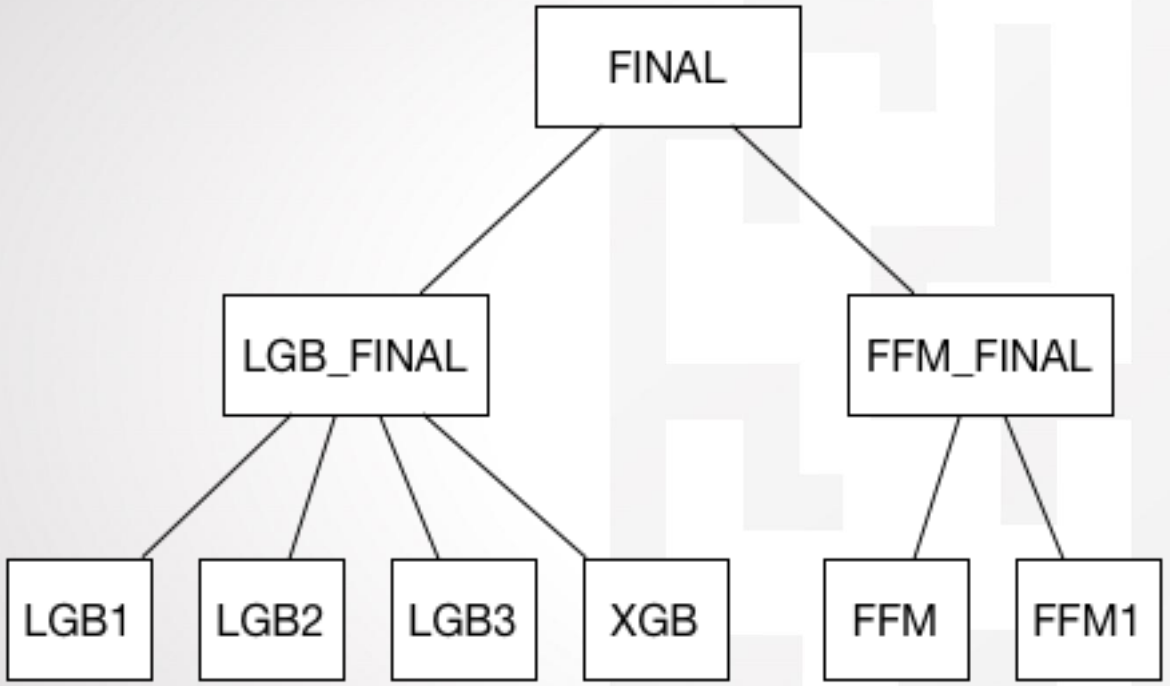


特征	分裂点	产生特征
Age	20，30	[0~20],[21~29],[30~]
AppRate	0.5	[0~0.5],[0.5~]

样本	Age	AppRate	原特征	新特征
1	10	0.4	0：10 1：0.4	0：0：1 1：3：1
2	20	0.6	0：20 1：0.6	0：1：1 1：4：1
3	30	0.4	0：30 1：0.4	0：2：1 1：3：1

03 模型算法

模型结果/Model results



我们最终的线上成绩使用了LightGBM、Xgboost、Libffm3种模型，其中不同的LGB、FFM代表不同的模型参数，最后采用如左图的方式进行加权融合得到最终结果。

模型	B榜成绩
LightGBM	0.10198
LightGBMs+XGBoosts	0.101859
FFM	0.102359
LightGBMs+XGBoosts+FFMs	0.101341

04 总结

团队合作/Team work

Git

我们使用Bitbucket合作，共享维护代码

头脑风暴

每周一起头脑风暴，构造新特征实现并测试

团队合作

自动化测试脚本

使用自动化特征脚本，非常方便进行特征的测试和筛选

开黑

等待测试结果的同时可以开一把王者荣耀压压惊

数据处理/Data process

- 我们线下的特征提取及模型测试均是在32G内存的服务器上运行
  - 最终线上的LightGBM和Xgboost模型使用64G内存的服务器运行
- 特征提取使用Python的numpy、pandas及Map Reduce
- Tips
  - 使用shell脚本并行提取特征
  - 使用python的multiprocessing库可以加速特征的提取
  - 使用numpy.savez及scipy.csr_matrix完成特征文件的持久化
  - 。。。

感谢/Thanks

- 感谢队友们一起的努力，不到最后一刻不要放弃
- 感谢群里各位大佬们的无私奉献
- 感谢主办方

腾讯社交广告
Tencent Social Ads