2018科大讯飞AI营销算法大赛top19

——算法说明

1. 训练数据采用的是初赛的’round1\_iflyad\_train.txt’与复赛的’round2\_iflyad\_train.txt’两个数据集的结合。
2. 数据清洗：对训练集和测试集的空缺值用-1来填充
3. 特征提取：特征分为分类特征和数据特征两部分

### 模型1(Lgb1)：yubase\_stack.py

### 分类特征包括：

### 简单的bool类别['creative\_is\_download','creative\_has\_deeplink','os\_name','creative\_is\_jump'] 用0,1编码。

### ['city', 'province', 'make', 'model', 'osv', 'os\_name', 'adid', 'advert\_id', 'orderid','advert\_industry\_inner','campaign\_id', 'creative\_id', 'app\_cate\_id','app\_id', 'inner\_slot\_id', 'advert\_name', 'f\_channel', 'creative\_tp\_dnf']

### 以上分类特征进行了简单的编码 (LabelEncoder)

### 数据特征包括：时间特征和历史点击特征，其中时间特征直接对时间戳进行了处理，历史点击率计算了['advert\_id','advert\_industry\_inner','advert\_name','campaign\_id','creative\_height','creative\_tp\_dnf', 'creative\_width', 'province', 'f\_channel']的历史点击率。

### 模型选择与训练：对特征进行了用lightgbm模型五折交叉训练，得到的模型预测测试集。

模型2(Lgb2)：mybaseline\_kdxf\_stacking.py

### 分类特征包括：

### ['city','adid','advert\_id','advert\_industry\_inner\_1','advert\_industry\_inner\_2','creative\_type','campaign\_id','creative\_id','orderid','app\_cate\_id','app\_id','creative\_tp\_dnf','creative\_has\_deeplink','province','inner\_slot\_id','advert\_name','f\_channel','model','os\_osv\_num','carrier','devtype','nnt','creative\_is\_download','make','os\_name','creative\_is\_jump']

### 对分类特征进行了简单的编码 (LabelEncoder)

### 其中'advert\_industry\_inner\_1','advert\_industry\_inner\_2'是'advert\_industry\_inner'的\_的简单分离，'os\_osv\_num'是‘osv’与‘os’的组合特征，'adid'和'city'分别取了点击率较高的前3位和前

6位。

数据特征包括：

['creative\_height','creative\_width','hour','day']其中,'hour','day'是对‘time’时间戳处理后的特征

对'user\_tags'做了CountVector处理

### 模型选择与训练：对特征进行了用lightgbm模型五折交叉训练，得到的模型预测测试集。

### 模型3(xgb):xgb\_stack\_lgb\_df.py

### 分类特征包括：

### ['city','adid','advert\_id','advert\_industry\_inner\_1','advert\_industry\_inner\_2','creative\_type','campaign\_id','creative\_id','orderid','app\_cate\_id','app\_id','creative\_tp\_dnf','creative\_has\_deeplink','province','inner\_slot\_id','advert\_name','f\_channel','model','os\_osv\_num','carrier','devtype','nnt','creative\_is\_download','make','os\_name','creative\_is\_jump']

### 对分类特征进行了简单的编码 (LabelEncoder)

### 其中'advert\_industry\_inner\_1','advert\_industry\_inner\_2'是'advert\_industry\_inner'的\_的简单分离，'os\_osv\_num'是‘osv’与‘os’的组合特征，'adid'和'city'分别取了点击率较高的前3位和前

6位。

数据特征包括：

['creative\_height','creative\_width','hour','day']其中,'hour','day'是对‘time’时间戳处理后的特征

对'user\_tags'做了CountVector处理

### 模型选择与训练：把模型1与模型2的预测结果（'kdxf\_pro\_fea\_10-11-18-47.csv'，'yu\_pro\_fea\_10-14-21-39.csv'）加权（6:4）作为新的概率特征与模型3的特征输入xgboost进行训练，得到的模型预测测试集。

最终模型的融合：  
xgb\_stack\_lgb\_df.py得到的概率结果xgb\_lgb\_baseline.csv与mybaseline\_kdxf\_stacking.py得到的概率结果kdxf\_baseline.csv融合：  
0.5\*xgb\_lgb\_baseline+0.5\*kdxf\_baseline

优点：

(1)模型2特征处理了部分特征与点击率的关系，得出点击率较高的某类特征的前几位数影响很大（adid,city）。

(2)训练xgb模型前加入了两个lgb模型的加权预测结果作为新的特征，输入到模型中，结果有较大提升。

缺点：

1. 对user\_tags做的处理分析不够，仅仅用了CountVector
2. 特征之间的某些联系理解不够，没有深入作特征之间的交叉特征等。