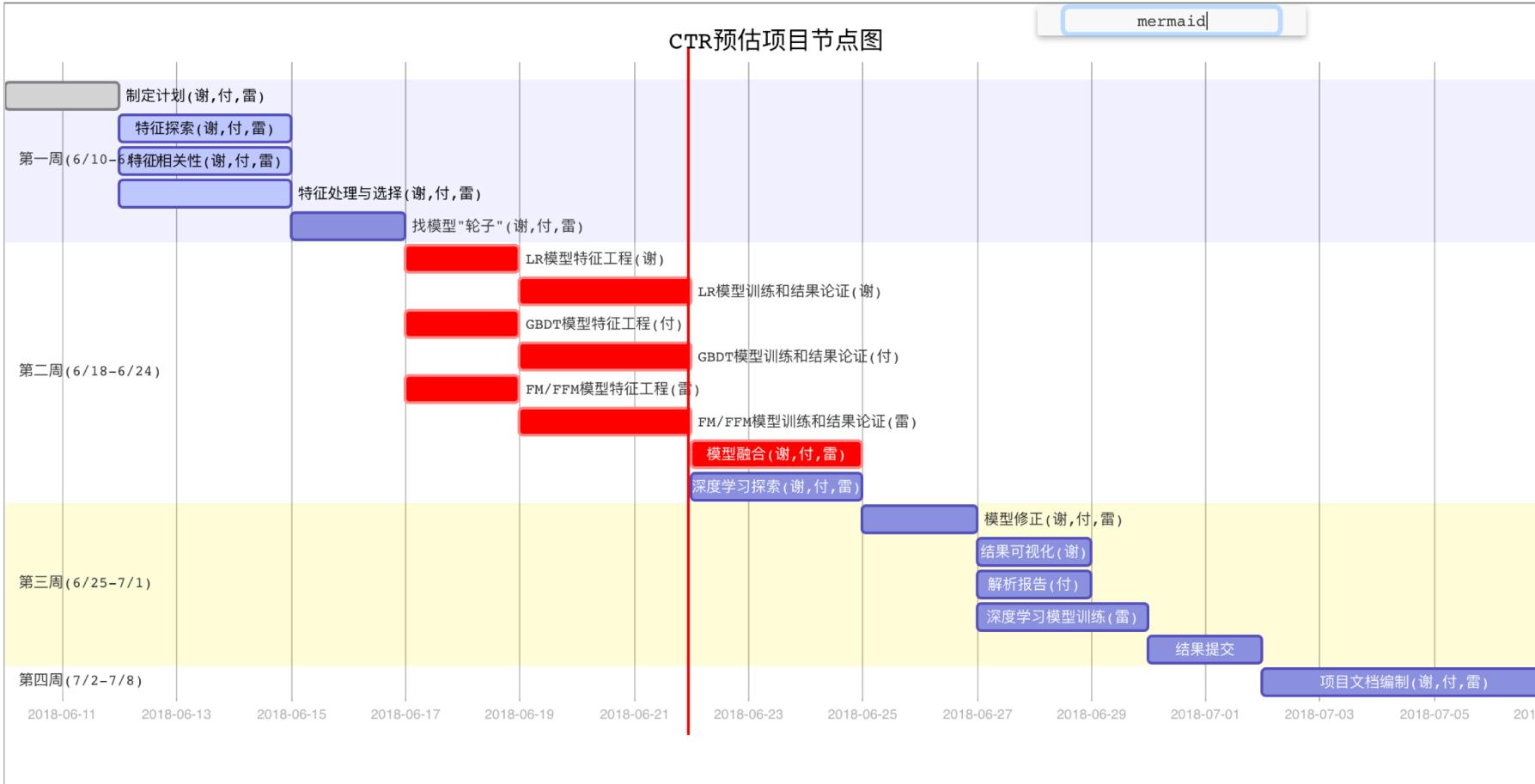


CTR 预估 项目汇报

第二周

成员：雷坤，付雄，谢飞

本周任务



单一模型

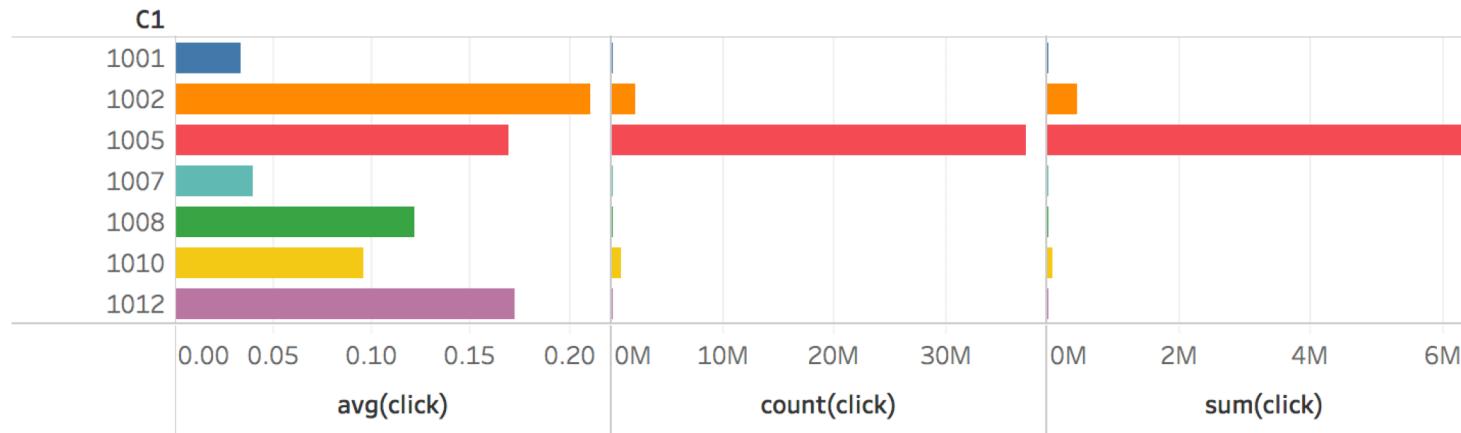
- LR模型
- Xgboost模型
- FM模型

LR 模型

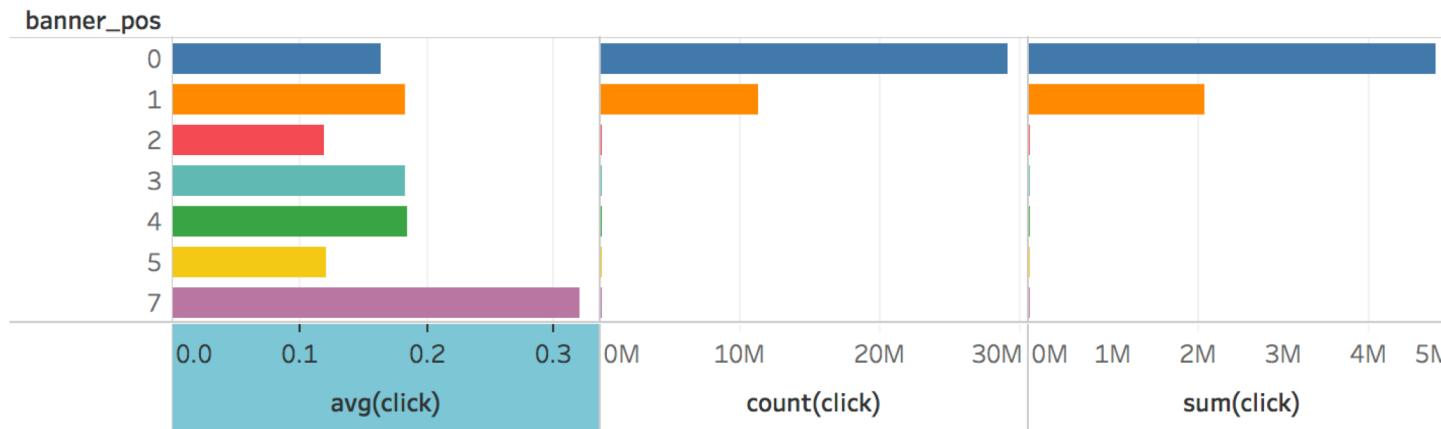
- 错误的CTR rate 编码
- 平均数编码

点击率的特殊性

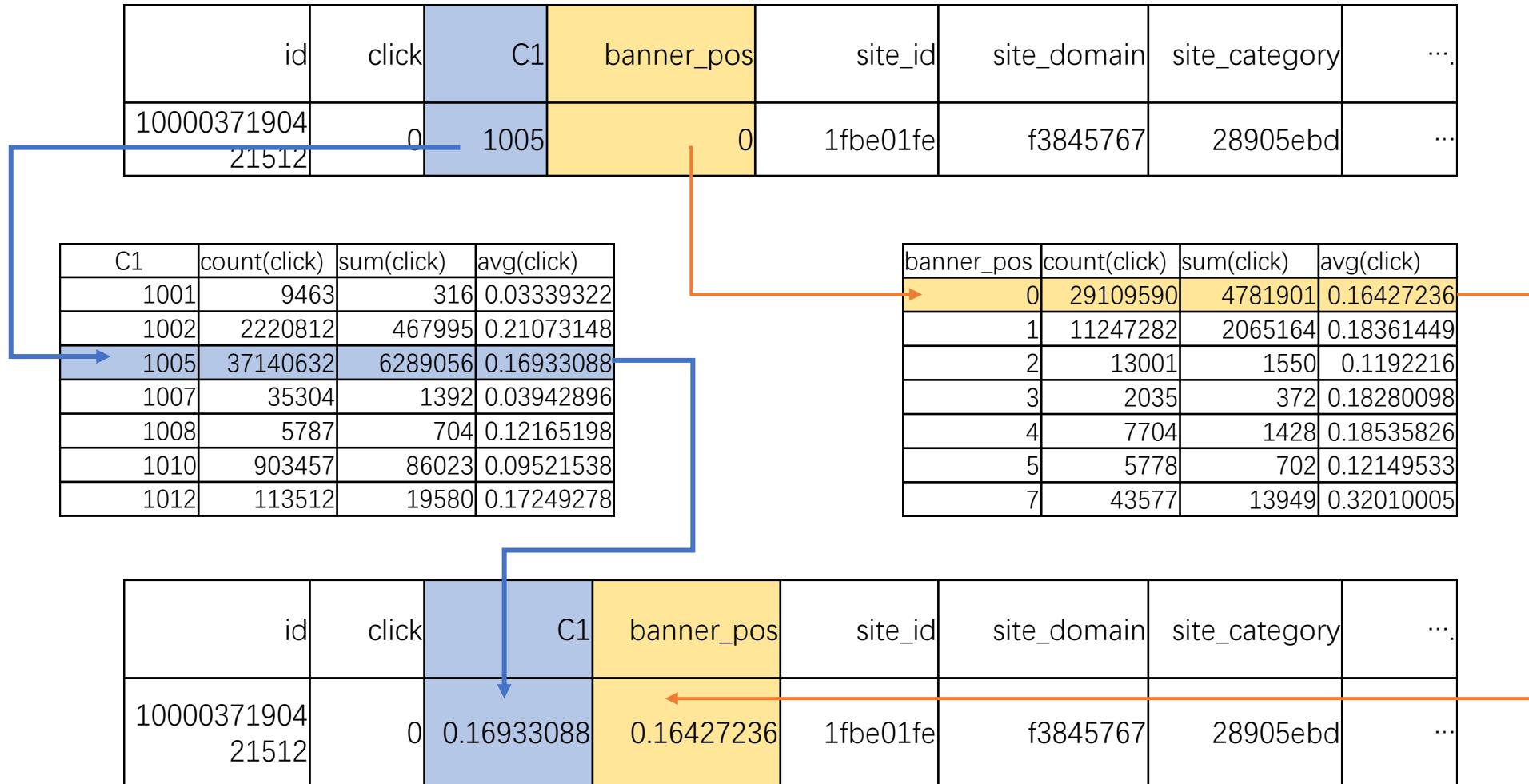
C1 vs Click



Banner pos vs Click



自创的编码



GridSearchCV

- penalty = ['l1','l2']
- Cs = [0.001, 0.01, 0.1]
- 最好的参数组合以及评分是：
- 'C': 0.01, 'penalty': 'l1'
- Loss = 0.311881

跑偏的尝试

- Leaderboard score:

Submission	Private Score	Public Score
1	0.5317915	0.5324564
2	0.4579208	0.4599311
3	0.4634562	0.4654680

跑偏的尝试

- PolyNomial

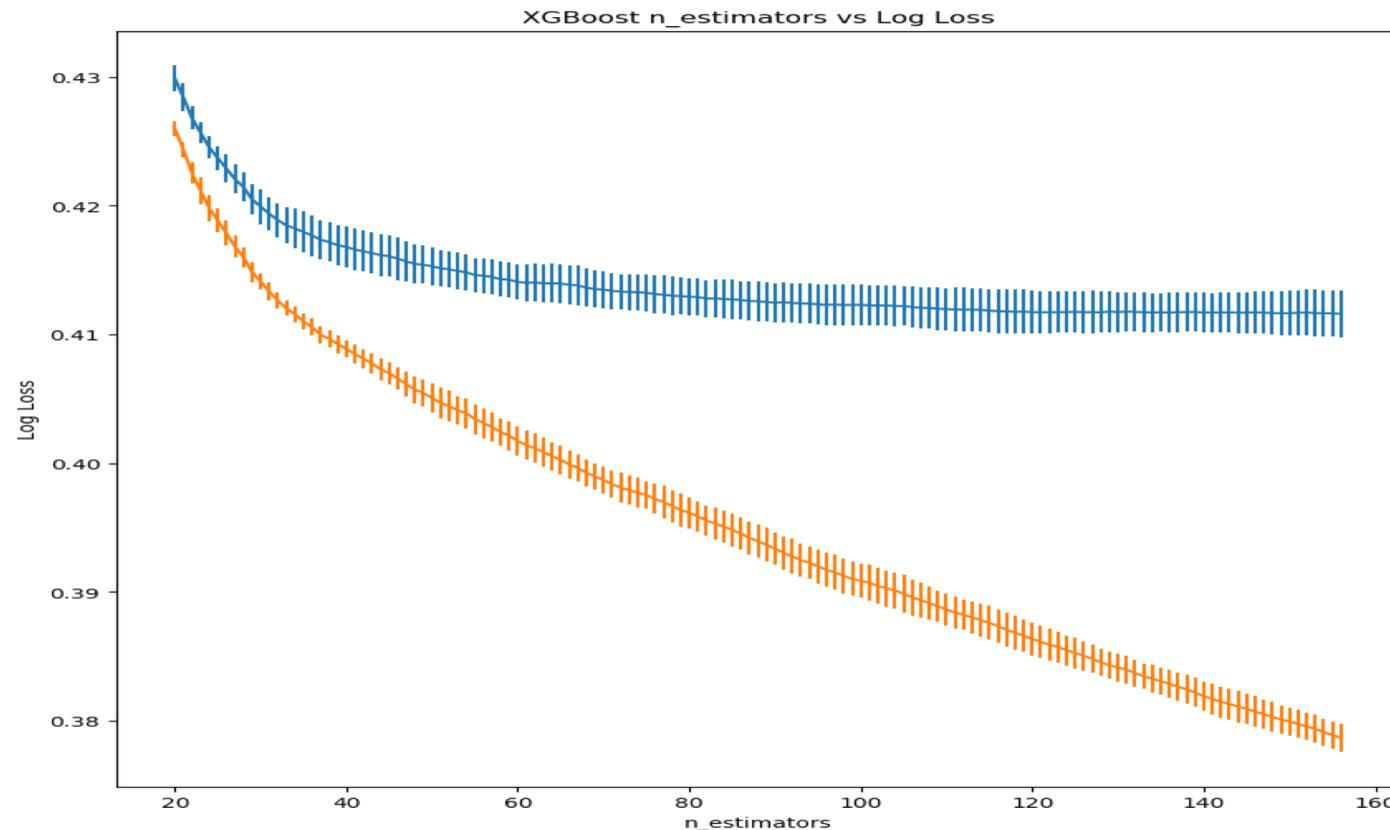
Submission	Private Score	Public Score
1	0.5317915	0.5324564
2	0.4579208	0.4599311
3	0.4634562	0.4654680

Xgboost模型

- 编码：哈希编码
- Hour = hour_int + day_week + hour_day
- 增加feature：User = device_ip + device_id

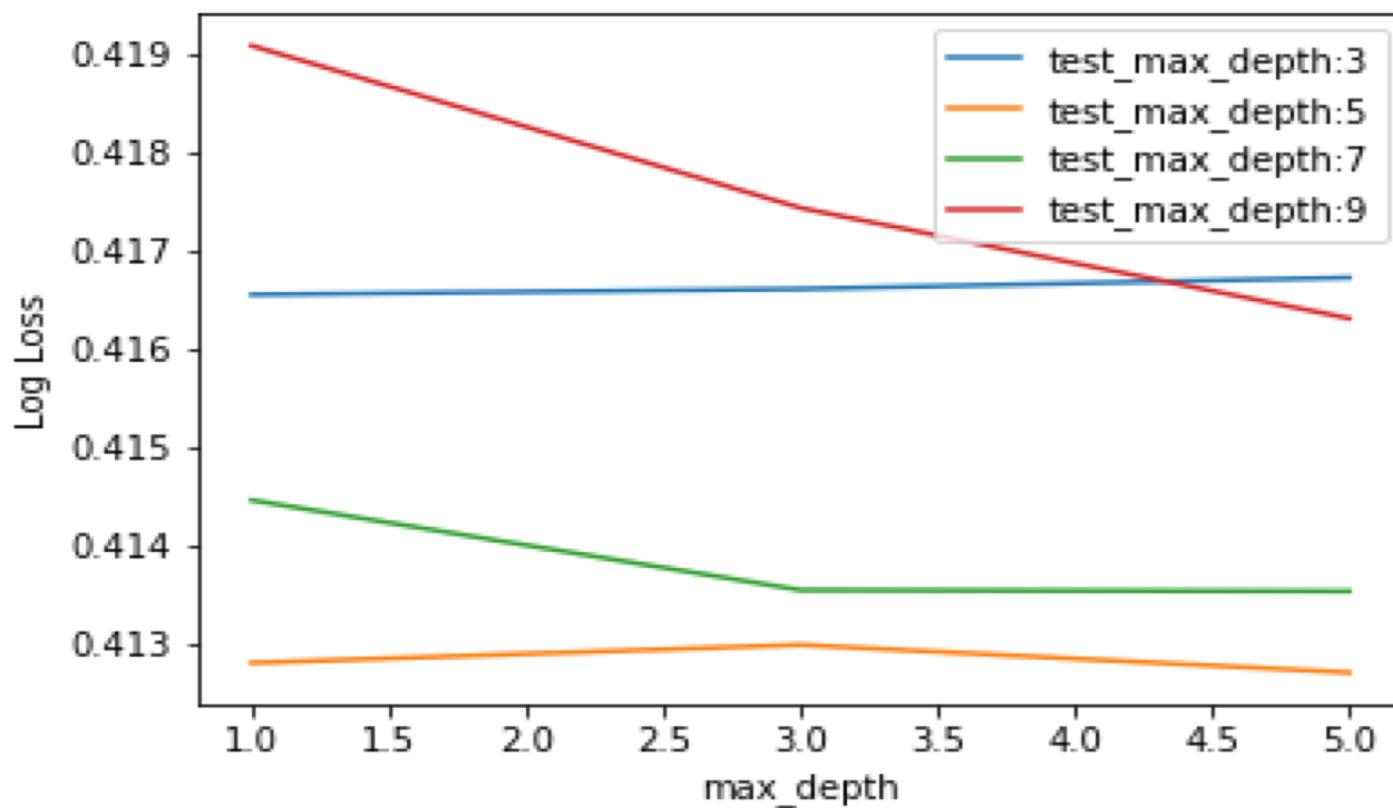
Xgboost 调参

- 1. cv寻找最佳的参数n_estimators



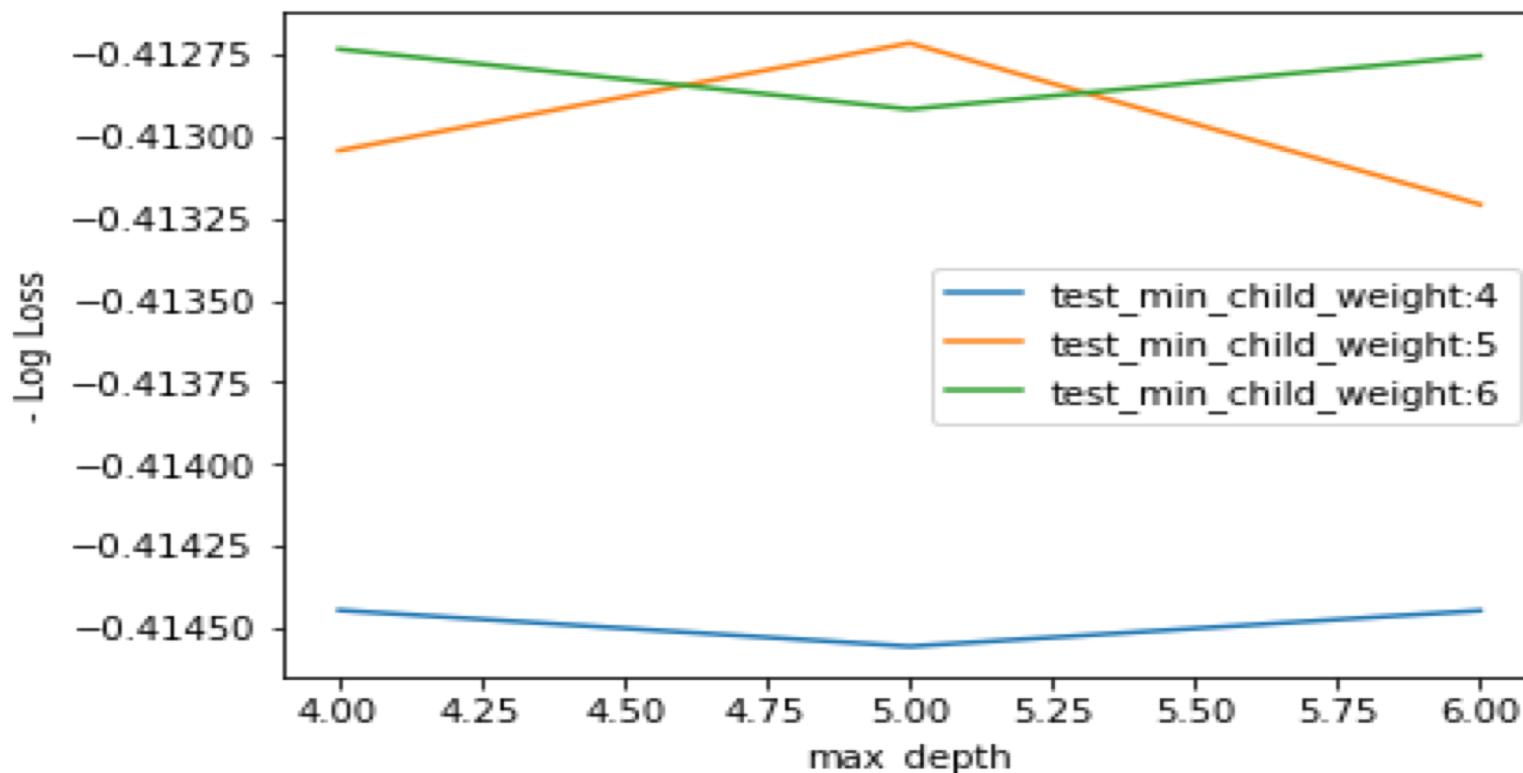
Xgboost 调参

- 2. 调整树的参数 : max_depth & min_child_weight
 - 步长为2



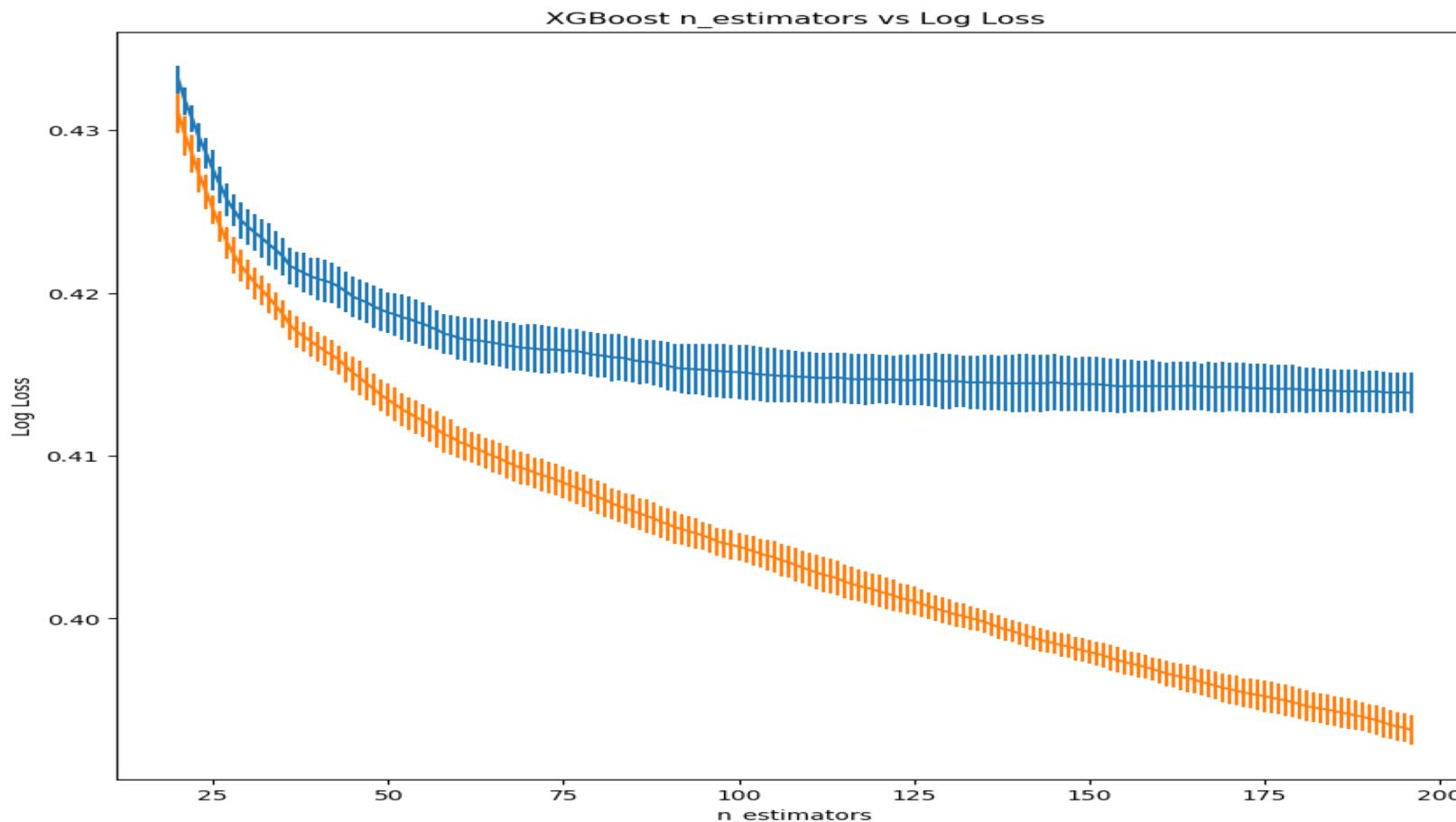
Xgboost 调参

- 3. 调整树的参数 : max_depth & min_child_weight
 - 步长为1



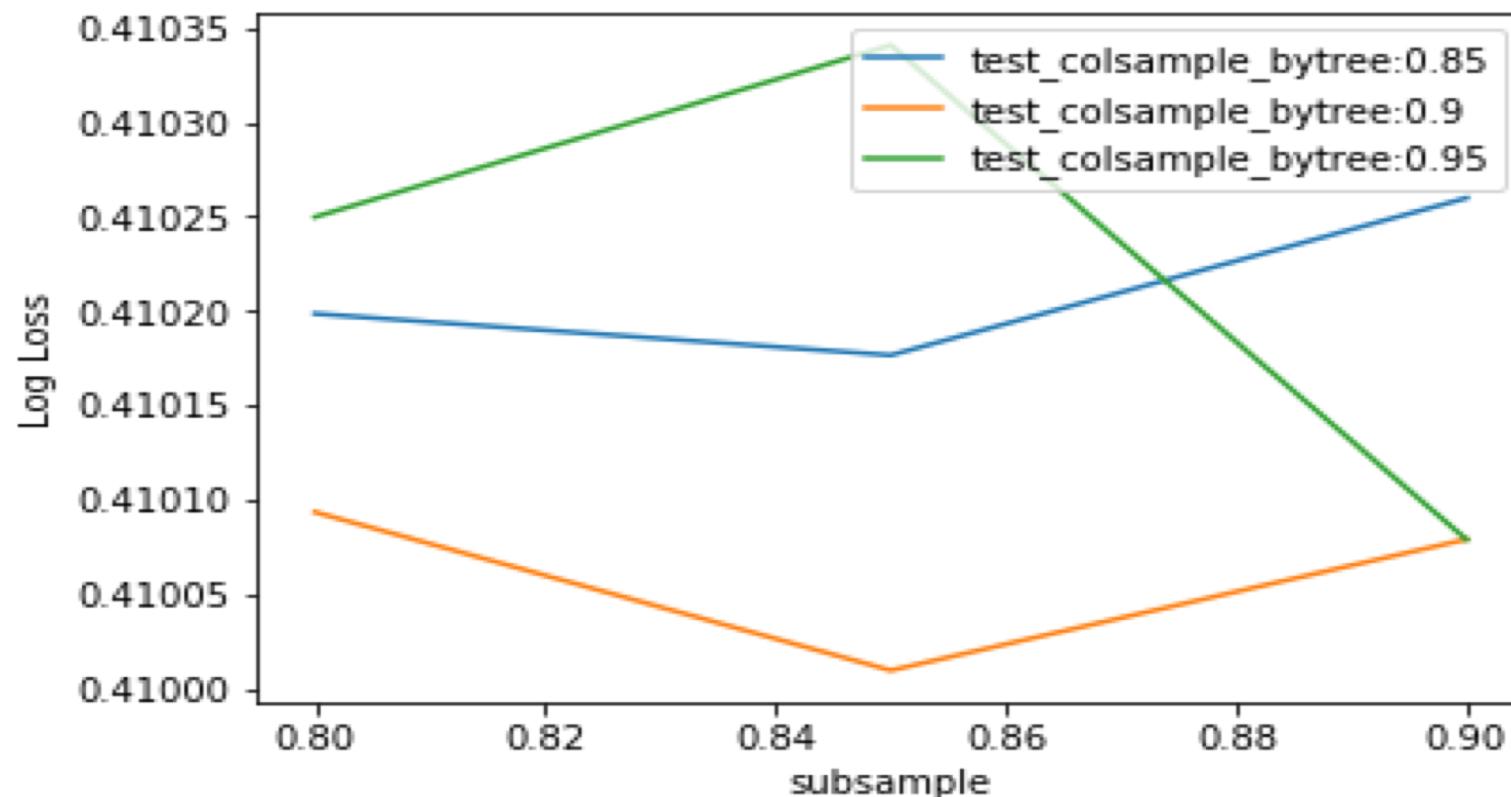
Xgboost 调参

- 4. 再次调整弱分类器数目



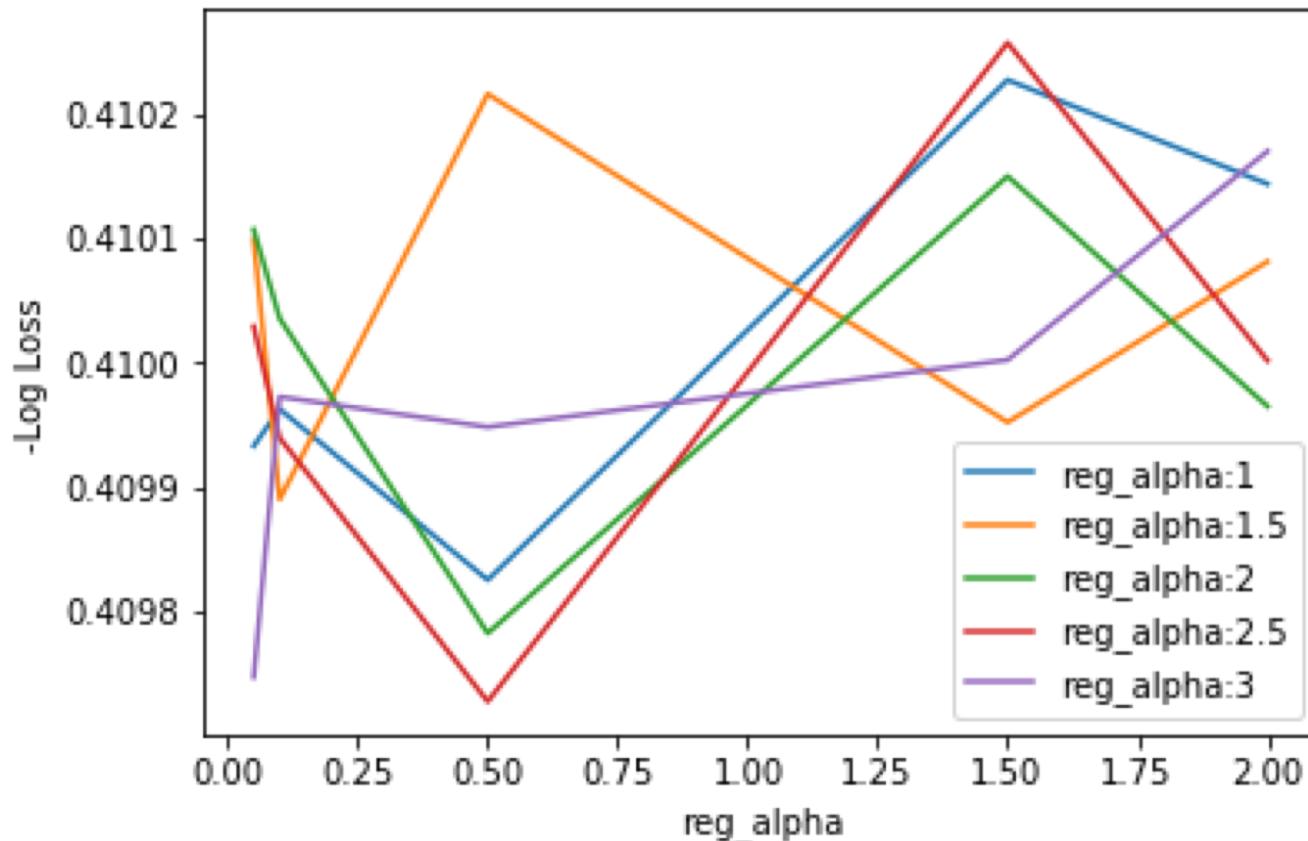
Xgboost 调参

- 5. 调整树的参数：subsample 和 colsample_bytree



Xgboost 调参

- 6.调整正则化参数：reg_alpha 和 reg_lambda



Xgboost 调参

- 7. 调整学习率 learning_rate = 0.05
- 最终最优参数：
 - 'learning_rate': 0.05,
 - 'max_depth' : 5
 - 'min_child_weight' : 5
 - n_estimators': 529, 'subsample': 0.85}
- logloss of train is: 0.3872155

Xgboost 表现

- Leaderboard score:

Submission	Private Score	Public Score
1	0.4031177	0.4047974

FFM模型

FM模型的方程为:

$$y(\mathbf{x}) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j \quad (1)$$

而FFM模型的方程为:

$$y(\mathbf{x}) = w_0 + \sum_{i=1}^n w_i x_i + \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{v}_{i,f_j}, \mathbf{v}_{j,f_i} \rangle x_i x_j \quad (2)$$

FFM模型

- 注意事项：
 - 样本归一化
 - 特征归一化
 - 省略零值特征
 - FFM需要设定early stopping

FFM模型 • 特征处理一

```
# 剔除特征app_id,site_id,site_domain,c14
# 对剩下的特征分为3类进行不同的处理:

# 对特征取值中计数大于10的进行独热编码,小于10次的统统归为一起,独热编码
count_category_features = ['app_domain',
                            'device_model',
                            'C17', 'C20']

# 对特征的每个取值进行独热编码
each_category_features = ['C1', 'banner_pos',
                           'site_category',
                           'app_category',
                           'device_type', 'device_conn_type',
                           'C15', 'C16', 'C18', 'C19', 'C21'
                           ]

# 连续值,对取值不进行独热编码,而是填入数值
continuous_value_features = ['hour', #填入点击率
                             'device_id', 'device_ip' #填入点击率,测试集中点击率等于0
                             ]
```

FFM模型 • 特征处理二

```
# 不剔除特征app_id,site_id,site_domain,c14

# 对特征取值中计数大于10的进行独热编码,小于10次的统统归为一起,独热编码
count_category_features = ['site_id', 'site_domain',
                            'app_id', 'app_domain',
                            'device_model',
                            'C14', 'C17', 'C20'
                           ]

# 对特征的每个取值进行独热编码
each_category_features = ['C1', 'banner_pos',
                           'site_category',
                           'app_category',
                           'device_type', 'device_conn_type',
                           'C15', 'C16', 'C18', 'C19', 'C21'
                          ]

# 连续值,对取值不进行独热编码,而是填入数值
continuous_value_features = ['hour', #填入点击率
                             'device_id', 'device_ip' #填入点击率,测试集中点击率等于0
                            ]
```

FFM模型 • 特征处理三

```
# 不剔除特征app_id,site_id,site_domain,c14

# 对特征取值中计数大于10的进行独热编码,小于10次的统统归为一起,独热编码
count_category_features = ['site_id','site_domain',
                            'app_id','app_domain',
                            'device_model',
                            'C14','C17', 'C20'
                           ]

# 对特征的每个取值进行独热编码
each_category_features = ['C1','banner_pos',
                           'site_category',
                           'app_category',
                           'device_type','device_conn_type',
                           'C15', 'C16', 'C18', 'C19','C21'
                          ]

# 连续值,对取值不进行独热编码,而是填入数值
continuous_value_features = ['hour',#填入点击率
                             'device_id','device_ip' #填入点击率,测试集中点击率等于0
                            ]
```

FFM模型 • 模型训练

数据样集

将train.csv按照日期141021~141030分成了10份数据集，并且这10份数据集进行了随机打乱。

模型的训练集train_sample200w.csv从141021~1410229这9份数据集中随机挑选5.6%的数据，最终整合为一个大约200w的随机训练样集。

从141030这份数据集中随机抽取1/40的数据，分别当作验证样集validate_sample10w.csv和测试样集test_sample10w.csv。

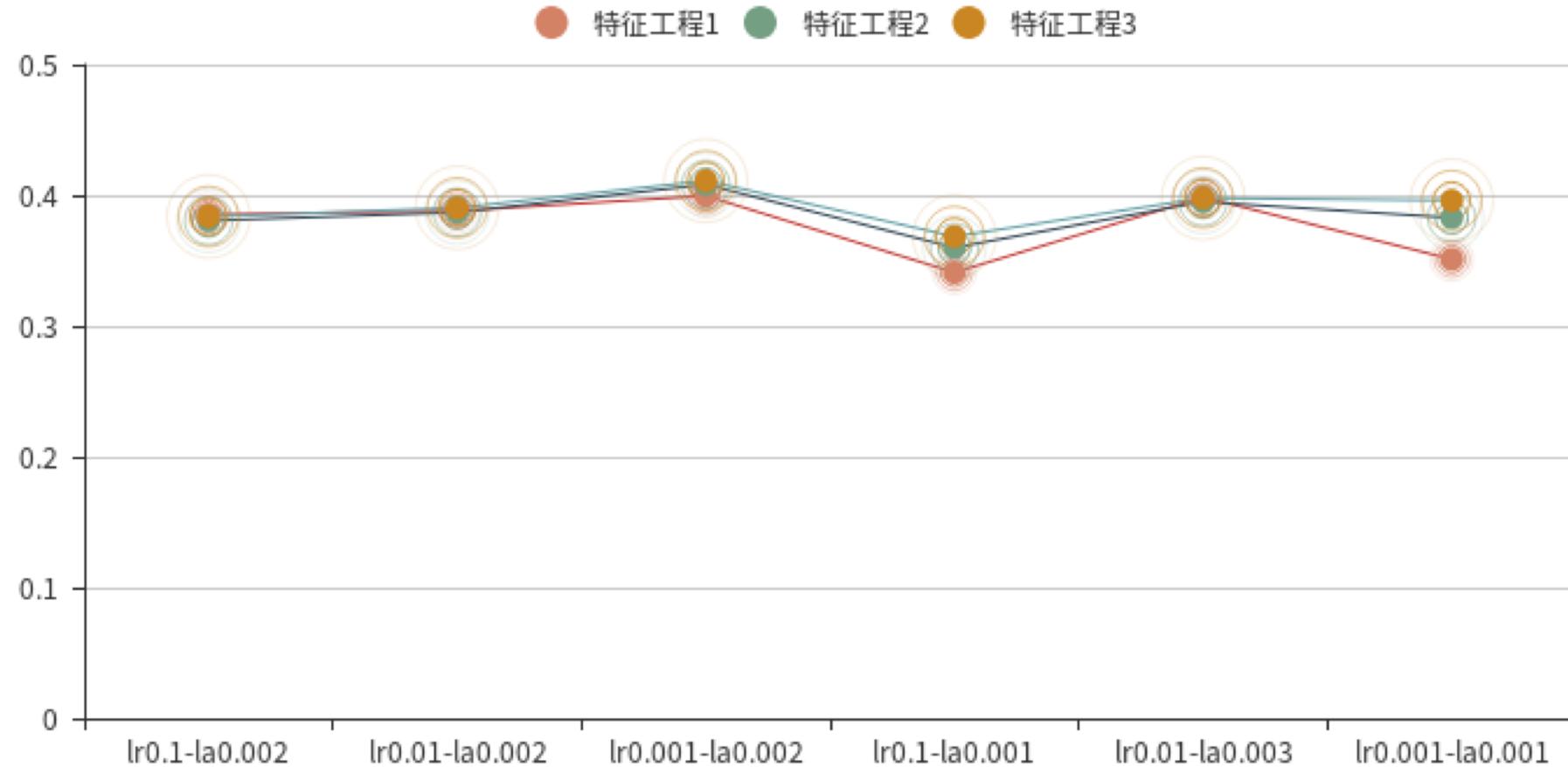
FFM模型 • 模型训练

超参数假定

- 采用的优化方法为adagrad
- 学习率lr分别设定了0.001,0.01,0.1三个取值
- 采用L2正则，正则化参数分别设定了0.002,0.001,0.003三个取值

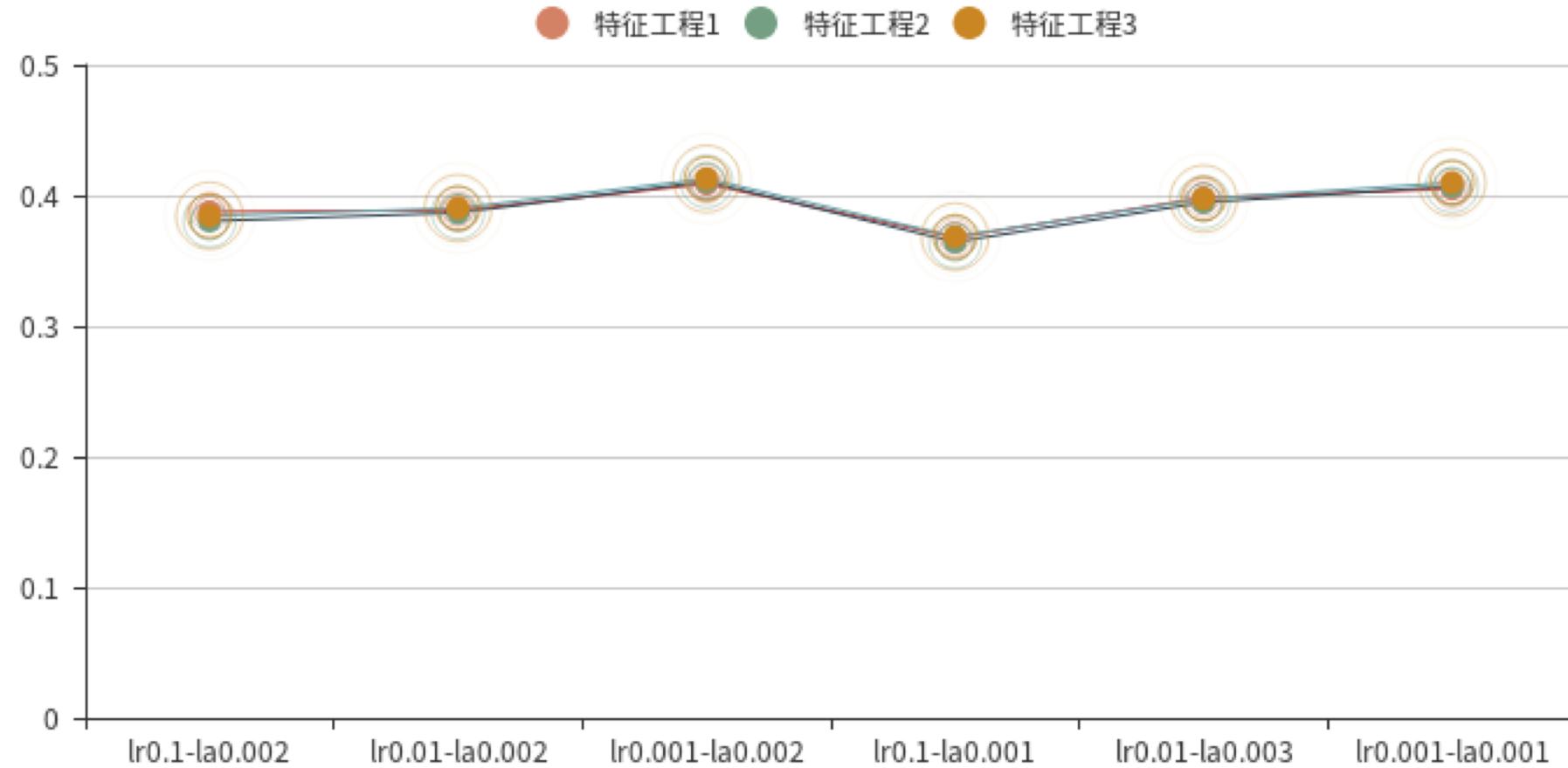
FFM模型 • 结果分析

FFM模型训练结果，logloss



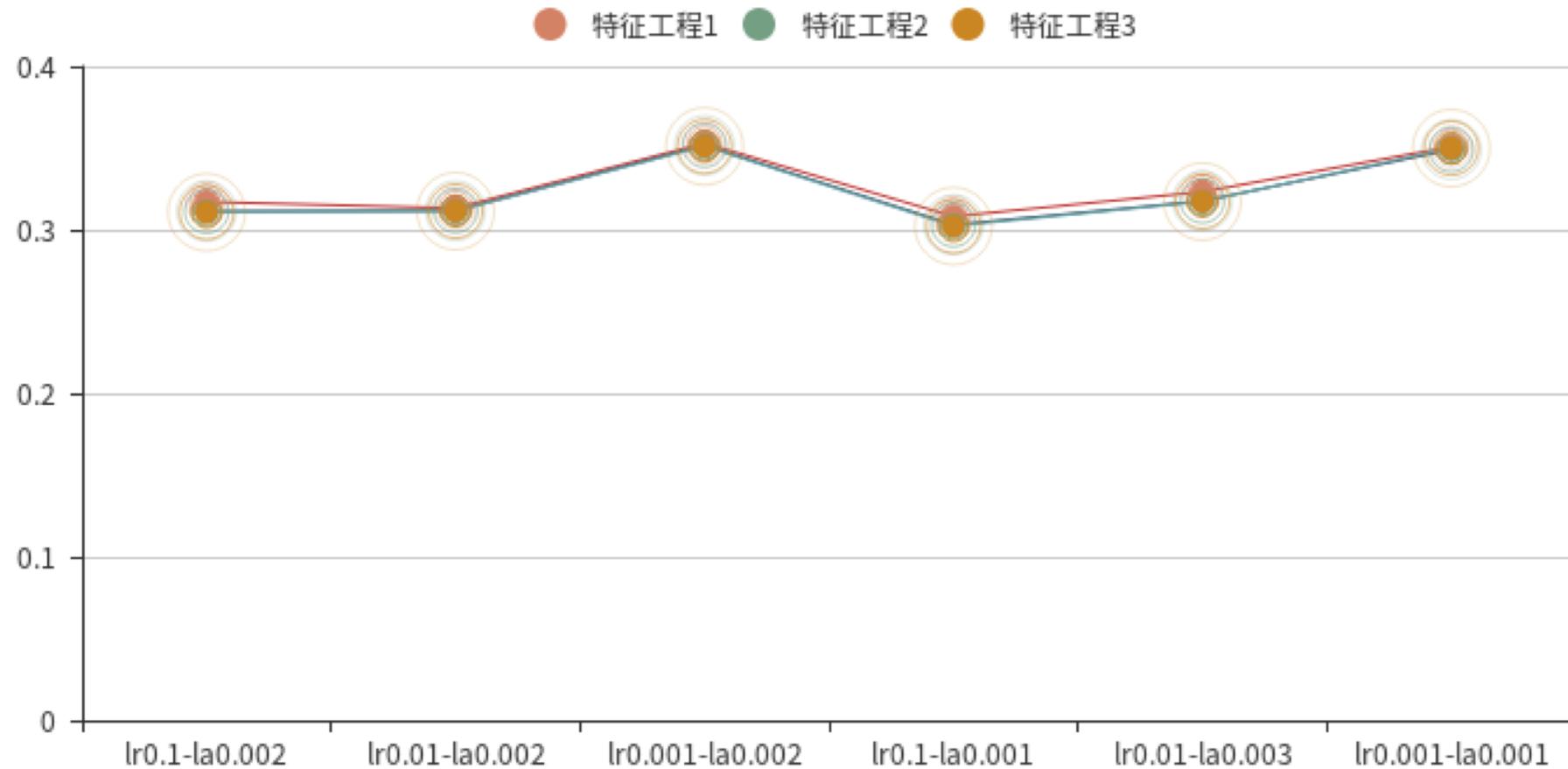
FFM模型 • 模型对比

FM模型训练结果，logloss



FFM模型 • 模型对比

LR模型训练结果，logloss



FFM 表现

- Leaderboard score:

Submission	Private Score	Public Score
1	0.4516173	0.45363119
2	0.4068746	0.408488850

总结—坑

- 错误的编码
- 轮子的选择
- 训练集与测试集的取值
- 模型的理解

下阶段的工作

- 继续探索特征工程
- LR->FTRL
- 模型融合
- 深度学习模型的探索 : DeepFMM