**第一名**

我们解题步骤分为数据预处理、数据去噪、特征提取、模型构建和模型融合5个部分。由于在特征上提出了稀疏特征的转化率编码方案，在模型上提出了nffm系列模型，并解决了一系列模型训练以及模型实现问题，我们才能取得好的成绩。特征抽取分为4个部分，即转化率、点击特征、安装特征和时间特征。模型上，我们采用了4种不同的模型，包括一个传统的GBDT模型，以及另外三个深度学习模型，分别是wide&deep网络、pnn网络和nffm网络。GBDT模型我们使用的是lightgbm，而wide&deep网络、pnn网络和nffm网络都是我们使用tensorflow和tflearn自己实现的，也正基于这点，使得我们很容易做模型改进、微调，这是非常有趣的事情。

**第二名**

首先是复赛数据量较大的问题。我想这也是所有参赛队伍都要面对的。由于数据量较大，所以我们对训练集进行了筛选，最后选用的是26-30的数据为训练集，且对30号的数据进行了预处理，过滤掉平均转化回流时间较长的app。另外，我们使用的训练验证集的划分方式是五折交叉验证的方式。虽然训练集筛选掉了一大半的数据，但是五折交叉验证跑26-30号数据的全集仍需要6小时以上（特征维度50+），所以我们在换b榜之前经常选用随机采样的方法来验证特征的有效性，采样比例是0.3。这样能最大限度的提高工作效率。

第二点，关于特征的提取，相信很多队伍也遇到过瓶颈。事实上，我们选择的方式是多思考赛题的实际意义。赛题要求是预测广告被点击后发生激活的概率，而用户点击广告后是否激活的一个主要驱动力就是用户对广告是否有需求，以及用户自身的安装力，所以用户的点击、安装历史以及用户针对具体某个App的点击安装历史尤为重要。基于这样的思考，我们构造出了一系列效果显著的特征，包括用户前一段时间内是否有点击、安装的记录；用户-App对是否在之前出现过、出现过多少次；用户-App对出现的次数等。此外，广告的位置信息也尤为重要。不同的广告位受到的关注度不同，所取得的推广效果也不同。

基于此，我们挖掘了数个与广告位相关的特征，包括转化率特征、历史点击量特征，以及用户、App与广告位的一些**组合特征**。此外，很多人都注意到了转化率特征的重要性，但很多人容易忽略的一点是用户的转化率特征。需要注意的是在利用用户转化率时需要进行恰当的分类，以免出现拟合的现象。

我们的算法框架为：数据清洗、数据划分、特征提取、模型训练、模型融合。

数据清洗应该算是我们的一个亮点了。由于转化回流时间有长有短，所以最后五天的label可能是不准确的，尤其是第30天。如果将第30天的数据全部删除，将会丢失掉大量有用信息；如果全部保留，又引入了相当程度的噪声。而我们发现，转化回流时间是与App ID有关的。于是我们统计了每个App ID的平均转化回流时间，并且删除掉了第30天中平均转化回流时间偏长的数据。这样处理之后，性能稍微有了一些提升。

此外，此前介绍，我们对特征也进行了充分的分析。对于position ID的充分挖掘、对于用户交互历史的充分分析都为我们带来了显著的提升效果。此外还有交叉特征的转化率在后期也起到了一定的效果。

数据划分部分，由于机器条件的限制，我们使用了第26天-第30天的数据。关于训练集验证集的划分，我们采用的是5折交叉验证。

特征提取部分，我们将其分为四项：

   1）基础特征，包括用户的基本特征、广告的基本特征、上下文特征；

   2）统计特征，对基础特征进行交叉后再统计，包括count操作和unique操作；

   3） 时间相关特征，主要统计了用户或用户-App在前一段时间内的点击次数或者安装次数；

   4）概率估计特征，对很多ID类特征，包括交叉ID类特征做了概率估计。

模型训练部分，我们刚开始一直使用的是LightGBM，训练速度非常快，在验证特征有效性方面可以大大缩短时间。我们的模型融合采用的是Stacking方法。除了LightGBM之外，我们又训练了FFM、LR、GBDT、ET模型。最终Stacking帮助我们提高了2.5个万分点左右。

**第三名**

比赛的题目是广告转化率问题，我们将它转化成了二分类问题。我们定义了App非当天转化率，并在删除错误样本比例和保留样本比例中找到了平衡点，充分使用了数据。在特征提取中我们会根据业务理解以及自身的经验提取特征，并对每个特征做分析，比如特征间的相关性，特征不同取值下转化率是否有明显变化。我们使用了xgb，lgb，ffm作为最后的模型，在xgb，lgb模型中我们使用了连续特征和类别特征训练，而ffm模型中我们使用xgb模型的输出作为ffm模型的输入，最后将三个模型通stacking的方法融合。