用户预订售卖房型概率预测

“到极限了吗？”团队解决方案

目录

团队介绍

团队成员：

谢玄 数据分析师 上海创冰信息科技有限公司

关乔 北京大学 计算机科学与技术 学生

雒航通 西安理工大学 自动化与信息技术 学生

最终成绩：A榜 0.470371 \ B榜 0.506046

比赛背景

调研表明，大部分用户除了对于酒店有偏好外，也有对于酒店房型的偏好。不同的酒店房型会提供酒店不同服务和礼惠政策等，这使得提供更多服务的同时，带来了用户一定程度的挑选时间。如何根据在用户的历史信息，挖掘出用户对于某些房型偏好，也为了节省用户的挑选时间和提供更好的服务。

比赛背景

首先就是问题的定性，这道题目的目标是判断用户对选择酒店的哪个房型进行订购。由于需要预测的目标取值为{0,1}，所以自然地将这一问题视作二分类的问题，并且建立模型预测出每一个售卖房型被预订的概率，同一orderid下概率最大的即为预测出的用户最终预订房型。

比赛背景

其次就是问题分析，用户选择怎样的房型受哪些因素的影响？

第一用户角度，用户的习惯和偏好应该会是一个很重要的影响因素。比如用户过去选择的酒店是不是这次选择的酒店，房型是不是这次选择的房型。

第二酒店角度，酒店的历史成交信息。比如一般酒店都会有受欢迎的房型，那么用户最终选择的可能性就会最大。

数据清洗

时间处理：经过统计，所给出的训练集的时间在2013-04-14到2013-04-20之间，而测试集的时间在2013-04-21到2013-04-27之间，我们把训练集和测试集的时间全部转化成0到6之间。

异常值处理：分析会发现有些数据上次预定的时间(orderdate_lastord)比本次预定时间(orderdate)还大，故在训练数据中把这部分数据剔除。剔除前：线下auc: 0.95543, 线上：0.502123，剔除后：线下auc: 0.9559, 线上：0.502843

数据清洗

冗余数据处理：数据中roomtag_6全部都为0，roomtag_6_lastord（也几乎全为0，极少部分为空），orderbehavior_4_ratio_1month, orderbehavior_5_ratio_1month和orderbehavior_3_ratio_1month这几列特征也全为空，我们也把这部分特征全部剔除掉，一方面也可以节省空间和时间，另一方面这一部分无用特征也可能会对模型学习有一些微弱的影响，这部分数据删除之后，结果也还是有很微弱的提升。

数据集划分

我们分别测试了通过随机划分数据集和按uid划分数据集2种方式，其实从直观上感觉随机划分数据集可能会把那些属于同一个uid的那些特征分开，直接随机划分数据集可能没有按uid划分效果好，通过实验也验证了我们的想法，我们最后采用的方式是按uid划分数据集，其中4/5的数据用于训练，1/5的数据用于测试，这样一方面既能保证训练数据有足够的多，另一方面也能保证有一定的数据用于验证，使模型不至于过拟合。提升效果：千分之五

特征工程

1、现在和过去的对比：这部分主要通过对比现在用户访问的酒店和用户以前的酒店订购行为的统计信息进行对比，反应出用户喜好和习惯。

2、每个订单内的酒店信息：比如通过订单分组统计出最小最大的价格，最小最大的优惠，数值数据的排序特征等。通过订单，物理房型分组统计最小最大的价格，最小最大的优惠，数值数据的排序特征等。

“ Applied machine learning is basically feature engineering.” -- Andrew Ng

特征工程

3、不同房型的订购统计信息：经过分析我们认为用户应该是先选择物理房型然后选择房型。通过统计一段时间内的用户对物理房型的的选择信息，和一段时间内的用户对不同房型的选择信息，来体现不同物理房型和不同房型的选择概率。

4、交叉特征：通过分析实际意义来对原始特征和统计特征做交叉运算，体现出数据的波动和变化。

“ Applied machine learning is basically feature engineering.” -- Andrew Ng

模型构建

因为本赛题数据量巨大，对于一般的算法考虑到运行的时间复杂度应该选择速度快并且应用广泛的模型。其次分类问题目前比较流行，并且泛化性能好，对非线性数据有很好学习能力的就是xgb和lgb模型，所以开始就选择了这两个模型。

前期：数据量相对小一直使用xgb模型，由于赛题是选择用户最可能选择的房型，是一个排序问题，所以在模型训练的时候使用auc来作为模型评价标准，并且线下编写测评函数进行验证保证同增减。

后期：随着特征数据的增大，发现xgb变得缓慢，反馈的时间变得很长，所以选择一边使用lgb快速训练模型，一遍训练xgb为后期的模型融合做准备。

模型融合

线性加权融合： p*xgb_score+q*lgb_score

概率相乘融合：（xgb_score^p）*(lgb_score^q)

其中xgb_score和lgb_score分别是xgb和lgb预测的用户选购房型的概率(概率在0到1之间)

结果：线性融合的效果较好，当p=0.2, q=0.8时线上效果最好，融合前最优单模型xgb线上分数为0.502789，lgb为0.505632，融合后达到了0.506046。

心得体会

关于比赛：

我们队伍由3个小伙伴组成，之前也都没什么经验，我们是在进入复赛前几天组队的，复赛开始后，我们迅速排到了第一名，并一直保持领先第2名超过2%，然后我们把队名改成了“到极限了吗”，由于保持领先优势太多再加上决赛结束前一段时间突然很忙，就没有花很多时间在上面，结果在比赛截止之前突然被现在的第一名反超，最后没有拿冠军，只拿到了第2名，有些遗憾，还是too young, too simple啊。

心得体会

关于特征工程：

根据个人的理解和相关经验，构造特征应该与实际生活经验相结合，如果不能想到一个特征对最后实际决策有任何影响，那就不要用，另一方面要多构造一些自己觉得对实际决策影响最大的相关特征。

最开始可能随便加很多特征结果都会有提高，但是越往后面发现再加入一些特征结果开始很难提高，甚至还可能下降(其实这个时候还远没有做到最好的程度)，因为前期加的太多不相关特征会影响模型的学习，因此对于这一点我个人的理解和建议是一方面不要随便乱加特征，另一方面就是每做到一定程度，就多测试一些，删除一些特征，事实上，个人认为剔除过拟合和不相关的特征是数据挖掘中最难的一个步骤，感觉也只能主要靠经验和多测试，虽然也有很多方法计算特征之间的相关性和相似度，但是这些方法实际运用时效果一般并不好。

心得体会

关于模型参数：

根据经验，200多维特征树的深度一般设置为5到6左右就够了，学习率一般越小结果会稍微好一些，但是当学习率很小之后就不一定了，但是学习率设小之后模型收敛一般会变慢一些，所以最开始可以把学习率设大一些，这样训练较快，比赛最后几天再把学习率设小一点(同时把迭代次数和early_stopping次数都变大).

心得体会

关于模型融合：

由于这个题目的数据量很大，模型的variance比较小，模型融合主要是减小variance，故在这个题目中模型融合应该影响比较小，事实上最后的实验也证实了我们的想法。

最后，建议大家多学习优秀的开源代码，多分析，多思考，多尝试。