题目:面向电信行业存量用户的智能套餐个性化匹配模型(2018 CCF-大数据竞赛(联通研究院举办))

网址：https://www.datafountain.cn/competitions/311/details

赛题背景：

电信产业作为国家基础产业之一，覆盖广、用户多，在支撑国家建设和发展方面尤为重要。随着互联网技术的快速发展和普及，用户消耗的流量也成井喷态势，近年来，电信运营商推出大量的电信套餐用以满足用户的差异化需求，面对种类繁多的套餐，如何选择最合适的一款对于运营商和用户来说都至关重要，尤其是在电信市场增速放缓，存量用户争夺愈发激烈的大背景下。针对电信套餐的个性化推荐问题，通过数据挖掘技术构建了基于用户消费行为的电信套餐个性化推荐模型，根据用户业务行为画像结果，分析出用户消费习惯及偏好，匹配用户最合适的套餐，提升用户感知，带动用户需求，从而达到用户价值提升的目标。
套餐的个性化推荐，能够在信息过载的环境中帮助用户发现合适套餐，也能将合适套餐信息推送给用户。解决的问题有两个：信息过载问题和用户无目的搜索问题。各种套餐满足了用户有明确目的时的主动查找需求，而个性化推荐能够在用户没有明确目的的时候帮助他们发现感兴趣的新内容。

赛题的任务(目的)：

此题利用已有的用户属性(如个人基本信息、用户画像信息等)、终端属性(如终端品牌等)、业务属性、消费习惯及偏好匹配用户最合适的套餐，对用户进行推送，完成后续个性化服务，是一个多分类任务。

数据集各个属性说明如下图所示：

字段	中文名	数据类型	说明
USERID	用户ID	VARCHAR2(50)	用户编码，标识用户的唯一字段
current_type	套餐	VARCHAR2(500)	/
service_type	套餐类型	VARCHAR2(10)	0：23G融合，1：2I2C，2：2G，3：3G，4：4G
is_mix_service	是否固移融合套餐	VARCHAR2(10)	1.是 0.否
online_time	在网时长	VARCHAR2(50)	/
1_total_fee	当月总出账金额_月	NUMBER	单位：元
2_total_fee	当月前1月总出账金额_月	NUMBER	单位：元
3_total_fee	当月前2月总出账金额_月	NUMBER	单位：元
4_total_fee	当月前3月总出账金额_月	NUMBER	单位：元
month_traffic	当月累计-流量	NUMBER	单位：MB
many_over_bill	连续超套	VARCHAR2(500)	1-是，0-否
contract_type	合约类型	VARCHAR2(500)	ZBG_DIM.DIM_CBSS_ACTIVITY_TYPE

contract_time	合约时长	VARCHAR2(500)	/
is_promise_low_consume	是否承诺低消用户	VARCHAR2(500)	1.是 0.否
net_service	网络口径用户	VARCHAR2(500)	20AAAAAA-2G
pay_times	交费次数	NUMBER	单位：次
pay_num	交费金额	NUMBER	单位：元
last_month_traffic	上月结转流量	NUMBER	单位：MB
local_trafffic_month	月累计-本地数据流量	NUMBER	单位：MB
local_caller_time	本地语音主叫通话时长	NUMBER	单位：分钟
service1_caller_time	套外主叫通话时长	NUMBER	单位：分钟
service2_caller_time	Service2_caller_time	NUMBER	单位：分钟
gender	性别	varchar2(100)	01.男 02女
age	年龄	varchar2(100)	/
complaint_level	投诉重要性	VARCHAR2 (1000)	1：普通，2：重要，3：重大
former_complaint_num	交费金历史投诉总量	NUMBER	单位：次
former_complaint_fee	历史执行补救费用交费金额	NUMBER	单位：分

Q:这里为什么要设置 AB 榜?
A:因为防止参赛选手模型过拟合，因此用于 AB 榜的测试数据有点不同，这样来验证模型或者算法的泛化性能，也可以理解为测试选手模型的稳定性。

主要思路：

在实际的操作过程中，主要分为数据处理、模型搭建、模型训练、优化结果四部分。接下来就详细的说下每个部分在这次比赛中的应用以及为什么这样做的原因

图 1.Xgb_model_1 文件夹下的文件组成部分

图 2.lgb_model 文件夹下的文件组成部分

1. 数据处理：

   在数据处理这部分，主要分为以下几个部分：

   - **数据分析**

     这是数据处理的第一步，在没有做任何预处理的情况下，利用第三方库 matplotlib、numpy、pandas 来对原始数据集中各个特征分布进行可视化,这样做的目的主要是看一下给的原始训练集和测试集的各个特征分布情况，这样可以为接下来的其他数据处理工作提供借鉴。

   - **数据清洗**

     这一部分主要是对一些脏数据进行剔除或者用其他的数据来代替，所谓的脏数据就是出现了异常值，或者不符合逻辑的一些值，还有就是数据类型不对。

   - **数据采样**

     由于数据分布太密集，并且本身的数据之间的值差性不大，为了防止数据的过拟合，我选择了 1/4 下采样(就是每四个值计算它们的平均值，然后用平均值来代替四个值，这样原始的四个值就变成了一个值)的方法对数据进行采样。这个比赛的目的是为电信用户匹配电信套餐，是一个多分类的机器学习任务，由于举办方给的数据集中的套餐类别之间是及其不平衡的，如下图所示，(左边一列表示的是套餐，右边一列表示其在数据集中的数量)，所以我们需要使用欠采样技术对套餐数目最多的进行削减，对套餐数目最少的进行过采样，这样减少类别之间的不平衡。

```
90063345    287219
89950166    133224
89950167     73842
99999828     52939
90109916     38096
89950168     33462
99999827     32531
99999826     29054
90155946     22037
99999830     21236
99999825     20350
Name: current_service, dtype: int64
```

- **离散特征与连续特征的处理**

所谓的离散特征,也叫类别特征，例如 many_over_bill 这样取值要么是 0，要么是 1 的，我们称为类别特征。对于 1_total_fee 这样取值 1.3，4.5 我们成为连续特征。在数据科学竞赛中，原始数据集中都会包含类别特征和连续特征，对于类别特征，一般的处理的方式是 one_hot 编码，至于为什么要 one_hot 编码，查阅相关博客说，都是使用欧式距离来计算距离的(就是中学学的两点之间的距离公式就好)，使用 one_hot 编码之后，所有的特征之间的距离都是一样的为根号 2，这样使得计算特征之间的相似性会显得合理点(类似于使用余弦来计算词向量之间的相似性道理一样),这里其实还没有讲清楚，不懂再问我。对于连续特征，一般操作就是归一化(我们这里使用的是 0-1 归一化，公式为(x-mean)/std)，使得不在同一量纲上的数据处在同一量纲(量纲这么解释：就是特征 A 的取值最小可以到 0.1，最大可以到 1000，这样就使得同一特征里面的数据差距太大，这样不利于模型的收敛)。

PS:在这里补充一点：对于树模型，一般不用对离散特征进行 one_hot 编码，以及不用对连续特征进行归一化操作.对于参数模型来说，是一定要进行 one_hot 编码和归一化操作。

- **特征挖掘**

这里主要使用了第三方库 seaborn 和 matplotlib 画的一个热度图，通过这个热度图我们可以看出特征与特征之间的相关性，越接近-1，表示负相关性越强，越接近+1 表示正相关性越强。通过热度图，我们可以进一步发现一些比较有用的特征，在数据科学竞赛中，很少直接全部使用原始数据集中的特征的，需要我们利用已有的数据集，去发现特征之间的关系，然后挖掘一些其它特征，在这里，我们可以举个例子：比如 1_total_fee,2_total_fee,3_total_fee,4_total_fee 这四个特征分别表示当前月出账金额、当前月前 1 月出账金额、当前月前 2 月出账金额、当前月前 3 月出账金额，然后我们想到了取这四个特征的最大值，最小值作为新的特征，这里只是稍微讲了下思想，具体怎么做的，可以去看源码里面的 data_process.py 文件。至于更高级的数据挖掘工具和算法在这里没有用，期待下次的比赛尝试。同时，通过相关性的计算(这里使用的 pandas 里面的 corr()函数来计算的)，可以知道哪些特征与目标特征相关性高，哪些特征与目标特征(目标特征就是你要预测的那个特征，在这里是 current_service)相关性低，非目标特征之间的相关性如果过高，则可以考虑删除相应的非目标特征，这样做主要是防止特征冗余，进一步导致的模型过拟合。非目标特征与目标特征之间的相关性过高，则要保留这些非目标特征。

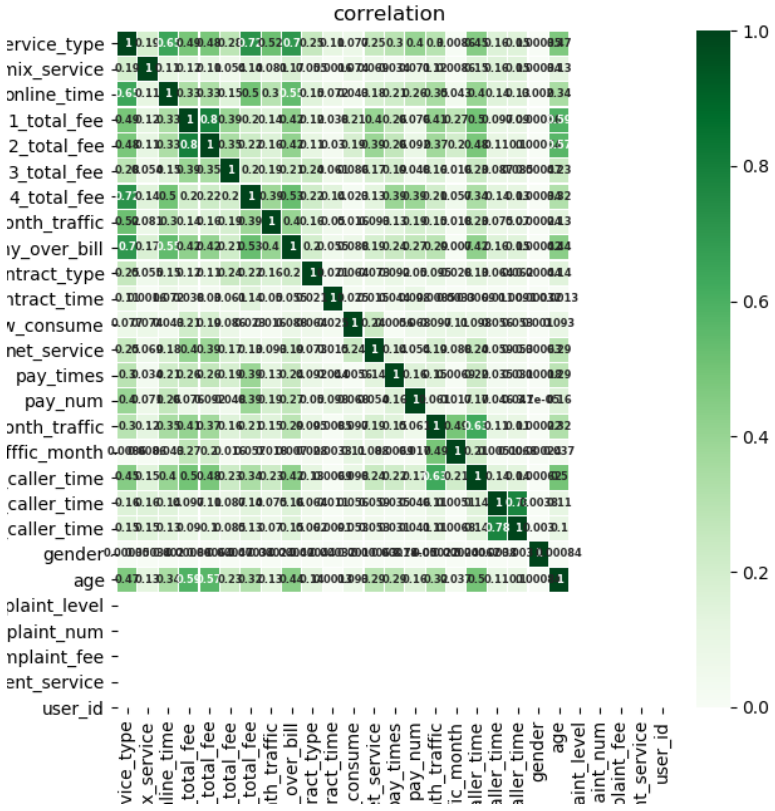


图 特征之间的相关性

- **特征重要性计算**

经过前面的特征处理之后，我们先利用随机森林、AdBoost、GBDT 对经过处理的数据集(加了特征之后的数据集)进行训练，然后通过 feature_importance()函数来计算所有特征的重要性(源代码里面都有)，所有特征重要性经过上述三个算法计算之后，可以做到心中有点底，就是知道哪些特征很重要，哪些特征不是特别重要。这样在后续的模型训练过程中，哪些可以去掉，哪些一定不能去掉。

- **降维**

主要是利用前面的技术对数据处理之后，根据已经掌握的的特征重要性，以及相关性，去掉某些贡献度较低特征，减少特征数量，加快模型的收敛速度以及防止模型过拟合。

2. 模型选择：

在这里我们选用了两个在比赛中用的比较多的基于树模型的算法，一个是 lightgbm(简称 lgb)，另外一个是 Xgboost(xgb),以下两个博客通俗的讲了下 xgb 和 lgb 的原理,这里可以参考以下博客，后面我会再详细说下这两个算法模型的原理和区别，以及怎么去用这两个算法：https://blog.csdn.net/github_38414650/article/details/76061893

https://blog.csdn.net/hqr20627/article/details/79426031

https://www.jianshu.com/p/48e82dbb142b

3. 参数调优：模型选好了，并且相应的运行代码大致框架搭建起来了(详细代码见 lgb.py 和 xgb_model.py)，那么接下来的事情就是如何给模型设置哪些参数，以及参数值的设置。接下来的工作就是调参，现在比较流行的调参方法主要有以下三种方式：贝叶斯调优、网格搜索调参(GridSearchCV)以及随机调参(RandomizedSearchCV)，在本次比赛中我们使用的是GridSearchCV，直接调用的是 sklearn.model_selection.GridSearchCV()这个函数。至于这三种调参方式有什么区别，以及如何使用，可参考：https://www.jianshu.com/p/5378ef009cae，调参的过程是很麻烦和辛苦的，需要我们一个一个的去调参，并不是一开始就把所有的参数输入到 GridSearchCV 中让它自己调参，而是每一次我们选一个参数，然后通过 GridSearchCV 来选择较好的值，然后下一步是在这一步的基础上，再对其他的参数进行较优值得选取。
4. 融合模型:
   在所有的工作都已经做的差不多的时候，我们通过训练集来训练我们搭建好的模型，然后再使用举办方给的测试集进行测试，最后提交结果，发现单模型的性能不是很好，这时候我们可以将两个单模型(xgb,lgb)进行融合，融合策略一般有以下几种:Stacking,Blending,对于多个提交的结果进行融合，在本次比赛中，我们使用的是从提交的结果文件中进行融合(也成为投票，具体的代码见 simple_vote.py),关于上述的三个融合策略，可以参考以下博客：
   https://blog.csdn.net/sinat_26917383/article/details/54667077

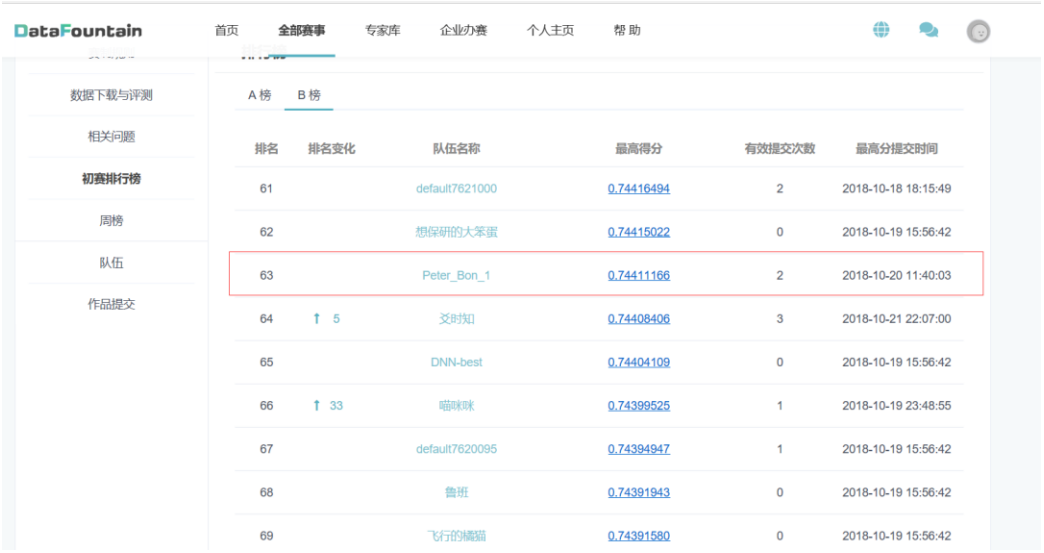
这就是这次比赛的一些实战经验，如果有什么疑问，可以和我说。

5. 结果提交：
   如果仅仅使用单模型的话
   lgb f1_score:0.7443
   xgb f1_score:0.750

融合后的结果：

初赛 A 榜：41/2546 f1_score:0.74688232

B 榜：63/2546 f1_score:0.74411166


DataFountain
首页
全部赛事
专家库
企业办赛
个人主页
帮助
数据下载与评测
相关问题
初赛排行榜
周榜
队伍
作品提交
A 榜
B 榜
排名 排名变化 队伍名称 最高得分 有效提交次数 最高分提交时间
61 default7621000 0.74416494 2 2018-10-18 18:15:49
62 想保研的大笨蛋 0.74415022 0 2018-10-19 15:56:42
63 Peter_Bon_1 0.74411166 2 2018-10-20 11:40:03
64 ↑ 5 艾时知 0.74408406 3 2018-10-21 22:07:00
65 DNN-best 0.74404109 0 2018-10-19 15:56:42
66 ↑ 33 喵咪咪 0.74399525 1 2018-10-19 23:48:55
67 default7620095 0.74394947 1 2018-10-19 15:56:42
68 鲁班 0.74391943 0 2018-10-19 15:56:42
69 飞行的橘猫 0.74391580 0 2018-10-19 15:56:42