马上 AI 全球挑战者大赛——违约用户风险预测

团队名称：大吉大利今晚吃鸡

队员：徐侃、林智敏、胡聪、陈涛

一、方案概述

在金融领域，无论是投资理财还是借贷放款，风险控制永远是业务的核心基础。马上消费金融举办的违约风险用户预测比赛就是希望能够借助人工智能和大数据等技术解决金融风控问题。官方提供了12w的训练数据，包括订单信息、商品信息、地址信息、在网购平台上的信用信息、以及用户的实名认证信息、银行卡信息等，以及用户逾期标志，需要我们去预测4w的预测数据，预测这4w用户的违约概率。显然这是个分类问题，明确了方向我们主要从数据预处理、特征工程和模型设计这三个方面入手。

由于在用户信息表里很多字段缺失率较高，我们依据用户该属性是否为空，将其转为0/1特征。然后统计每个样本中为零的个数，其实也就是缺失的个数，再除以用户信息表的总维度，计算得到用户信息的缺失率，构造评估用户资料完整度的特征。在特征工程中，我们采用逐一添加表的形式来不断增加特征，这样的好处是可以直观得到强特到底在哪张表里，直观感受分数增长的梯度情况。基于以上的数据和特征处理，我们依次采用单模型、加权融合、stacking以及随机扰动等方式进行建模。通过线上测试得到，加权融合在本次比赛中是最有效的。

二、数据洞察

1.根据生日提取年龄。原始文件提供了用户出生日期这个字段，不过字段的值有多重形式，包括“1995-04-04”这种标准格式以及“90 后”还有部分乱码数据。我们先将乱码数据去除，再将“90 后”等转为“1990-01-01”这样的标准格式，最后用 2017 去减得到用户年龄特征。

2.bank 表里的银行名称字段存在中英文混合。自己构建一个字典 ,统一将英文替换成中文，减少银行的类别,方便后续处理。

3.存在疑似用户多次贷款现象。在分析订单信息和地址信息时，存在两个贷款编号下的数据字段大量重复现象，因而怀疑是用一个人在两个时间点申请贷款所导致的，于是提取了用户的贷款次数。

三、特征工程

1. AUTH_INFO 表

1）身份证第一位 可以表示地区（1 表示华北，2 表示东北，3 表示华东）尝试将身份证信息转换为可以入模的变量；

2）认证时间的年、月、日、星期 将原始时间信息转化为可以入模的变量；

3）身份证是否存在 无法直接使用身份证信息，转换为是否存在身份证信息；类似的，我们对于认证时间、电话号码是否存在进行统计；

4）手机信息 认证数据中，针对电话号码，将号码前三位转换为对应运营商信息（移动、联通和电信）；

5）借贷时间减去认证时间 表示用户在认证后进行贷款申请的经过天数。对于欺诈用户，由于含有强烈的目的性，可能在认证后马上进行了借贷活动；而对于一般用户，可能在认证一段时间后才进行贷款申请。

2. CREDIT_INFO 表

经过分析，该部分数据来源于京东平台，主要包括信用评分、白条额度及使用额度。其中信用评分、白条都是京东通过分析用户购物、还款数据 ,所得到反应用户消费能力的分数。一般消费能力越强，其对应的分数越高。这里直接使用了该部分数据。同时，基于使用额度数据，将用户当前剩余额度及额度使用比例作为新特征。

3. USER_INFO 表

1）性别 映射到 0/1/2

2）QQ 及微信是否绑定 0/1

3）会员等级 从低到高，分别映射为 0,1，,，4

4）年龄 通过申请时间-生日年份来计算

5）统计缺失值个数 由于在该数据表中，存在大量的空字段，同时由于京东个人信息填写没有经过核验，真实性缺乏检验，这里没有继续将如行业、收入、学历等信息入模；转而采用统计空字段个数来表征该用户的个人信息维度。

4. BANK_INFO 表

1）分别统计了银行卡、储蓄卡、信用卡数量。

2）bankpred 银行进行 one-hot 编码，五折交叉用一层决策树对标签预测得到一列预测值，然后将预测概率作为特征放入模型。从而将 100 多维降为 1 维，从而避免将稀疏矩阵直接入模。

3）用户所在银行的违约率均值 先统计不同银行的违约率，然后可以得到一个用户的不同银行的违约率，取平均就好。

4）计算贷款次数 在分析订单信息和地址信息时，存在两个贷款编号下的数据字段大量重复现象，因而怀疑该贷款编号是用一个人在两个时间点申请贷款所导致的。于是提取了用户的贷款次数。

5. ORDER_INFO 表

该数据来源自用户订单信息，基于以下：

['amt_order','type_pay','time_order','phone','no_order_md5']字段进行去重，并进行统计

1）订单订单金额处理：分别统计总额、均值、标准差和 skew 值，订单金额越大，表明用户经济水平越好，其违约可能性越低。

2）预留不同的电话个数

3）订单时间信息处理：最大时间间隔，以及时间分别在上午、下午、晚上和凌晨的个数，在工作日、周末的数量，以及周几的众数；

4）product_id 为空的个数

5）最后一次消费时间和申请时间的间隔 时间交叉特征

6）缺失信息的比例

7）购买人的收货的地区 MD5 编码的 unique 除以总个数 确认他是给几个人买东西，能说明其经济实力

6. RECEIVE_INFO 表

1）统计 receive 表里面的地址的个数

2）receive 表里面的固话填写的频率

3）是否更换地区特征 如果是贷款之后容易违约的话，他购买的时候可能存在购买个数很少，更换地区，也能间接说明他生活状态不稳定。

四、特征预筛选

特征筛选，我们主要从三个方面着手：

1.删除缺失率高的特征 对于原始表里某些缺失比较高的特征，比如 user_info 表里的 hobby（缺失 87.46%）、marriage（缺失 92.61%）、income（缺失 92.36%）、degree（缺失 93.08%）、industry（缺失 93.14%）等，由于缺失较多，不可能直接用，但一开始又不想丢掉。所以，提取了该属性是否为空的特征，线上结果显示并没有用，于是果断删掉。

2.模型输出的特征重要性 我们使用的 LightGBM 模型会输出模型特征重要性，按照重

要性从高到低排序，将末尾几个重要性为 0 的删掉，线上确有小小提升。

3.根据线上反馈增删特征 之前将申请贷款时间转为 unix 时间直接入模，还提取了年、月特征。线下验证效果很爆炸，线上却是很萎靡，发生严重过拟合。想了想，时间直接入模本身就不合理，由于要预测的贷款时间都是 2017 年五月份，这就导致测试集得到的年月特征全部一样，毫无意义。

五、模型训练

在竞赛圈一般都是使用树模型，尤其以梯度提升树为典范，主流使用的是 XGBoost、LightGBM 和 CatBoost。由于 LightGBM 训练速度快，支持类别特征，且准确率也高，所以我们在单模型阶段大部分都是用的 LightGBM。

1.单模型

正如上文所述的，为了在特征工程处理时，保证特征的有效性和稳定性，我们采取较为保守的方式，固定使用 LightGBM 模型，并且保证模型参数的统一，模型参数调优是在确定最优特征之后进行的。

对于模型评估方面，我们线下采用两种验证方式：

（1）按照时间划分 由于要预测的是 2017 年 5 月的数据，所以我们将 4 月划为验证集，4 月之前为线下训练集，以 4 月的 AUC 成绩提升与否作为判断依据。

（2）5 折交叉验证 将训练集按照固定的正负样本比平均划分为五份，每份的数据量都和预测集的数据量接近，保证验证集合预测集有着相近的数据集分布，使评估结果更准确。

通过观察这两种验证方式的综合情况来决定是否使用该模型来提交。

2.加权融合

加权一直都是个玄学，单模型做到头了，那就试试融合吧。之前一度陷入了误区，将同样特征跑三个不同模型来进行加权，会提升但是效果不明显。我们最优单模型 0.83587

（LightGBM）与同特征的 XGBoost 和 CatBoost 加权后得到 0.83620，效果一般。通过翻看 github 和他人比赛经验，知道要突显特征差异性，于是找到队友之前的提交模型。我们之前是分开提取特征各自建模提交，思路不同导致各自提取的特征自然存在差异。将队友的 0.832 模型和我们最优模型加权，最终线上成绩 0.8386。

3.STACKING

我们试过两层的 STACKING，不过效果提升只有万分点，以大量的时间换取这点提升效果，显然是无法让我们满足，所以后面我们放弃了 STACKING,选择加权摸奖。

4.模型参数的随机扰动

这个思路来自于 Bryan 大神的参赛分享。采用 bagging 的思想 ,建立 20 个 LightGBM 模型，每个模型参数都是在哎最优参数的附近随机扰动，本来还可以对特征进行随机选取前 50，到前 100 这样的，但是由于我们特征维数只有 50+于是就不对其进行选取，最后对于这 20 个模型取平均。我们测试之后的提升效果也不好，弃用。

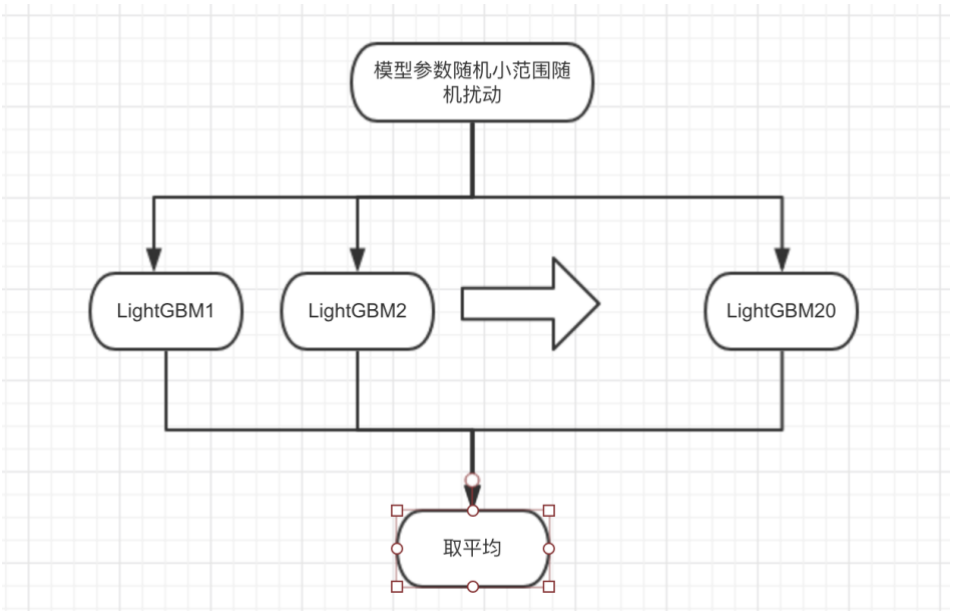

图 1 模型示意图

六、重要特征

列出模型所选的重要特征的前 20 个：表格样式如下：

表 1 特征重要度

特征名称	特征释义	特征重要性排名
credit_score	用户信用分	1
appl_auth_time	申请贷款时间减去认证时间	2
bankpred	对银行名称做的 onehot 然后五折交叉用决策树预测出一列	3
overtime_delta	统计各个省份的违约率	4
bank_rate_avg	一个用户所在银行的违约率的均值	5
count_rec_info	统计 receive 表里面的地址的次数	6
orderpred	order 表中支付方式和收货状态 横过来 五折交叉用决策树预测一列	7
atest_appl_order	申请贷款时间减去最后的 order 时间	8
order_null_avg	order 表中缺失值的均值	9
amt_order_sum	账单总额	10
null_coun	缺失值统计	11
amt_order_mean	每月账单均值	12
dur_day	order 表最大时间间隔	13
auth_time_day	认证时间的日期特征	14
amt_order_std	账单标准差	15
account_grade	用户账号等级	16
use_rate	额度使用率	17
night_order_count	晚上订单的统计	18
amt_order_skew	账单的斜度	19
id_len	Id 长度	20

七、创新点

1.对于缺失值的处理。不直接用缺失率较高的特征，也不直接抛弃，转为 01 特征，统计样本的缺失值个数，得到用户信息的完整度。

2.bankpred 特征的提取。将 bank 表里的 100 多个银行进行 one-hot,依次统计每个样本是否有该银行的卡，有则置 1，无则置 0，得到一个 100 多维的 01 稀疏阵，然后五折

交叉用决策树预测得到一列关于标签的预测值，这样就把 100 多维降成 1 维，再拿去给模型训练，防止直接用稀疏矩阵给模型带来影响。

3.特征差异性加权融合。让特征和模型都存在差异性，使得融合效果更上一层楼。

八、赛题思考

金融风控其实不仅仅是贷中的违约预测，还有贷前准入和贷后催收。如果能够对用户信息事先进行完备地审核，可以使用知识图谱的不一致性验证，这样可以大大降低风险。在贷中，不仅仅要对用户基本信息以及消费行为数据进行分析，还应该增加增加社交网络信息，多维度评估贷款人信用状况。同时贷后的催收是极为关键的一环，可以使用知识图谱挖掘用户的潜在联系人，即使用户故意隐身，提供的联系人也联系不上，我们可以自己去挖掘出他们社交关系，找到潜在联系人，大大提高催收率。