**马上AI全球挑战者大赛——违约用户风险预测**

## 一、方案概述

近年来，互联网金融已经是当今社会上的一个金融发展趋势。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。对于消费金融来说，其主要服务对象的特点是：额度小、人群大、周期短，这个特性导致其被公认为是风险最高的细分领域。

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。

基于本赛题，大数据金融的违约用户风险预测，本文解决方案具体包括以下步骤：

1.对用户的历史行为数据预处理操作；

2.根据历史行为划分训练集数据、验证集数据；

3.对用户历史数据进行特征工程操作；

4.对构建特征完成的样本集进行特征选择；

5.建立多个机器学习模型，并进行模型加权融合；

6.通过建立的模型，根据用户历史行为数据对用户在未来一个月是否会逾期还款进行预测。

其中，图1展示了基于大数据金融的违约用户风险预测解决方案的流程图。



**图1违约用户风险预测解决方案的流程图**

## 二、数据洞察

### 2.1 数据预处理

1.异常值处理：针对数据中存在未知的异常值，采取直接过滤的方法进行处理会减少训练样本数量，从这里出发，将异常值用-1或者其他有区别于特征正常值的数值进行填充；

2.缺失值的多维度处理：在征信领域，用户信息的完善程度可能会影响该用户的信用评级。一个信息完善程度为100%的用户比起完善程度为 50%的用户，会更加容易审核通过并得到借款。从这一点出发，对缺失值进行了多维度的分析和处理。按列（属性）统计缺失值个数，进一步得到各列的缺失比率，按对数据进行多维度处理，其中为数据集中某属性列缺失值个数，C为样本集总数，为数据集中该属性列缺失率：

3.其他处理：空格符处理，某些属性取值包含了空格字符，如“货到付款”和“货到付款 ”，它们明显是同一种取值，需要将空格符去除；城市名处理，包含有“重庆”、“重庆市”等取值，它们实际上是同一个城市，需要把字符中的“市”全部去掉。去掉类似于“市”的冗余之后，城市数目大大减少。

### 2.2 发现时序关系

根据用户历史数据，统计违约数量和未违约数量跟时间周期的关系，可视化实现如下图所示：

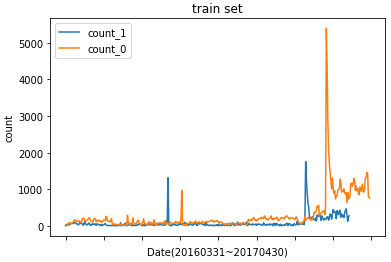
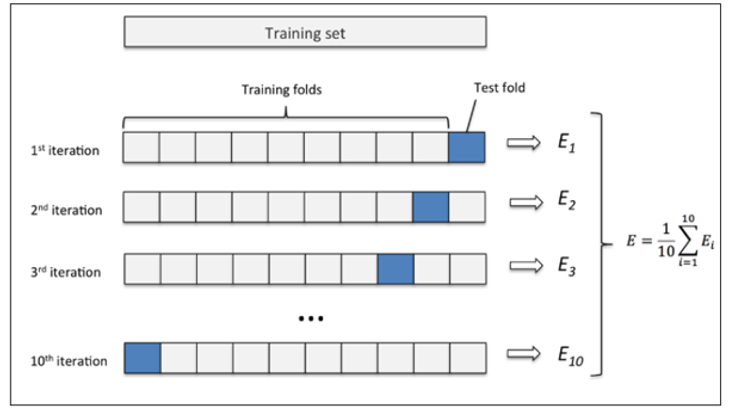


图2 违约数量和未违约数量跟时间周期的关系图

可以看出，时间对用户是否违约是成一定周期性的，且2017年明显比2016年的数量增加了很多，因此本文解决方案涉及很多时序特征。

### 2.3 划分训练集、验证集

对违约用户风险预测是一个长期且累积的过程，采取传统的按训练和测试集对应时间段滑窗法划分数据集并不是最佳方案，从这里出发，将历史用户数据全部用于训练集，更好的训练用户行为习惯，其中，验证集的构建采取交叉验证的方式，交叉验证如下图所示：



**图3 交叉验证示意图**

## 三、特征工程

**1、AUTH\_INFO用户认证表**

**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即增加一列有为1，无为0**；**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网购平台信用信息表**

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。

**1）网购平台信用额度减去使用额度** 表示用户剩余的使用额度

**3.USER\_INFO用户信息表**

**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** 通过申请贷款时间-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银行卡信息表**

**1）分别统计了银行卡、储蓄卡、信用卡数量。** 每个ID会有多个银行卡，利用groupby+agg统计每个ID的银行卡数量

**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** 每个ID会有多个不同电话

**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收货地址信息表**

**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。

### 3.2 信息完整度特征

主要基于auth、credit、user表提取，标记这三张表每条样本的信息完整度，定义为该条样本非空的属性数目/总属性数目。

### 3.3 one-hot特征

主要基于user表提取。

One-hot离散user表的sex、merriage、income、degree、qq\_bound、wechat\_bound、account\_grade属性。

### 3.4 业务特征

基于业务逻辑提取的特征，最有效的特征，主要基于credit、auth、bankcard、order表提取。

（1）用户贷款提交时间（applsbm\_time）和认证时间（auth\_time）之差

（2）用户贷款提交时间（applsbm\_time）和生日（birthday）之差(贷款时的年龄)

（3）信用评分（credit\_score）反序

（4）信用额度未使用值（quota减overdraft）

（5）信用额度使用比率（overdraft除以quota）

（6）信用额度使用值是否超过信用额度（overdraft是否大于quota）

（7）银行卡（bankname）数目（每个id会有多个银行卡）

（8）不同银行的银行卡（bankname）数目

（9）不同银行卡类型（card\_type）数目

（10）不同银行卡预留电话（phone）数目（有多个电话）

（11）统计每一行的缺失值占的个数，构造评估用户资料完整度的特征。

（11）提取order表的amt\_order次数、type\_pay\_在线支付、type\_pay——货到付款、sts\_order\_已完成次数，按id对order表去重，保留id重复的第一条样本

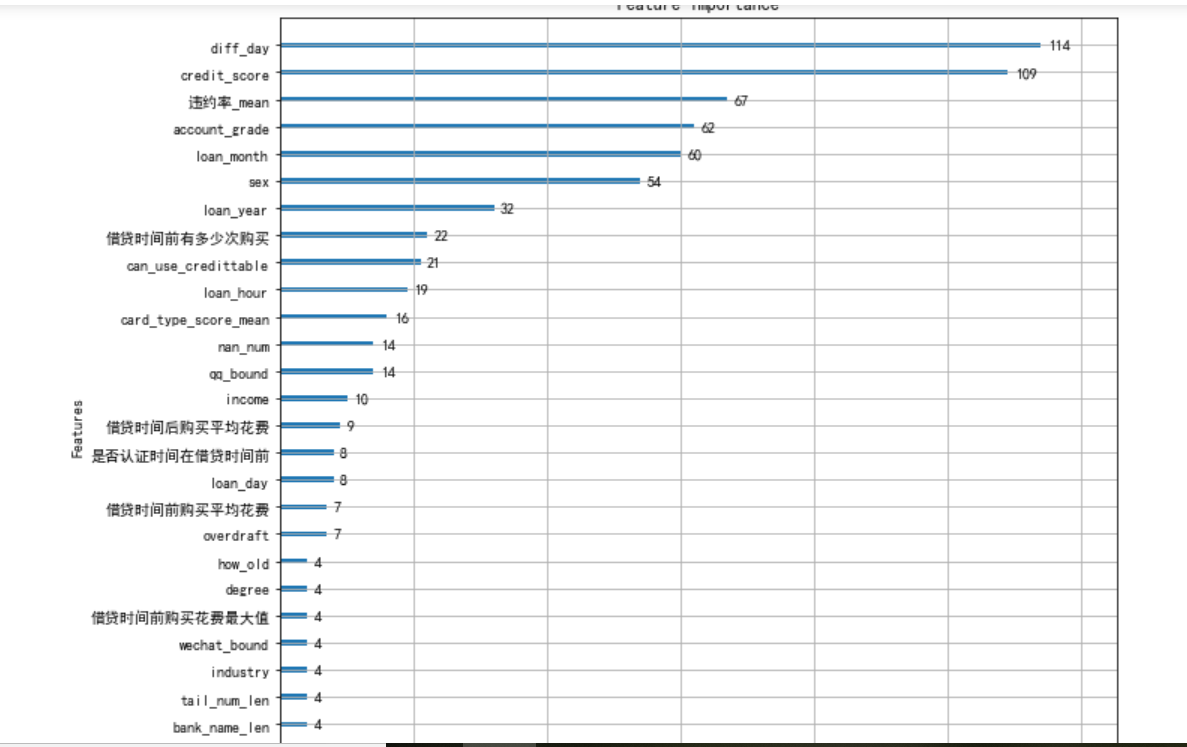
## 四、特征筛选

特征工程部分，构建了一系列基础特征、时序特征、业务特征、组合特征和离散特征等，所有特征加起来高达数百维，**高维特征一方面可能会导致维数灾难，另一方面很容易导致模型过拟合**。

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。从这一点出发，通过特征选择来降低特征维度。比较高效的是基于学习模型的特征排序方法，可以达到目的：模型学习的过程和特征选择的过程是同时进行的，因此我们采用这种方法，基于 xgboost 来做特征选择， xgboost 模型训练完成后可以feature\_importance函数输出特征的重要性，据此我们可以保留 top n 个特征，从而达到特征选择的目的。



**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，尤其以梯度提升为典范，**主流的一般都是 XGBoost、lightGBM、GBDT。本文共计四个xgb模型，分别进行参数扰动、特征扰动，单模型效果均通过调参和特征选择，保证单模型最优，按四个模型不同比例融合，最终生成模型结果。

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blending、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。

利用heamy库ModelsPipeline().Stack()函数进行特征融合，Regressor函数建立融合模型，roc\_auc\_score函数得分为0.81341，同时利用ModelsPipeline().blend()函数，特征得分0.81396.

**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XGBoost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。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，保证验证集和预测集有着相近的数据分布，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提交。

单模型XGBoost的AUC计算值为：0.8149717

**2、stacking与加权融合blending**

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。我的最优单模型XGBoost与同特征的LightBoost、GBDT加权后为0.8134186，stacking的结果为0.8139588，效果有微小提升，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，这里没有再进行特征工程。

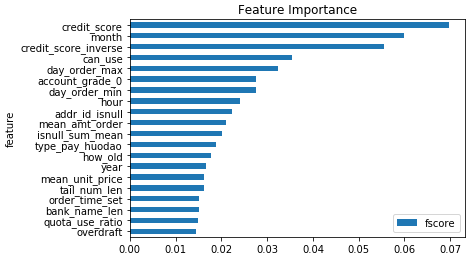
**4.模型参数的随机扰动**

采用bagging的思想，此处建立3个XGBoost模型，每个模型参数都是在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3个模型取平均。



## 六、重要特征

通过XGBOOST模型输出特征重要性，降序排序，选取top20，可视化如下：



**图4 特征重要性排序**

列出模型所选的重要特征的前20个：表格样式如下：

|  |  |  |
| --- | --- | --- |
| 特征名称 | 特征释义 | 特征重要性排名 |
| credit\_score | 网购平台信用评分 | 1 |
| month | 当前样本月份 | 2 |
| credit\_score\_inverse | 网购平台信用评分反序 | 3 |
| can\_use | 剩余使用贷款额 | 4 |
| day\_order\_max | 当前样本时间-订单最大时间 | 5 |
| account\_grade\_0 | 会员级别0类型的离散值 | 6 |
| day\_order\_min | 当前样本时间-订单最小时间 | 7 |
| hour | 当前样本小时数 | 8 |
| addr\_id\_isnull | 地址信息是否为null | 9 |
| mean\_amt\_order | 订单金额的均值 | 10 |
| isnull\_sum\_mean | 缺失值总数的均值 | 11 |
| type\_pay\_huodao | 货到付款类型的数量 | 12 |
| how\_old | 用户年龄 | 13 |
| year | 当前样本的年份(2016/2017) | 14 |
| mean\_unit\_price | 商品单价均值 | 15 |
| tail\_num\_len | 银行卡号码长度 | 16 |
| order\_time\_set | 用户下单时间不同的次数 | 17 |
| bank\_name\_len | 银行卡长度 | 18 |
| quota\_use\_ratio | 用户贷款额使用率 | 19 |
| overdraft | 网购平台信用额度使用值 | 20 |

## 七、创新点

### 7.1 特征

原始数据集很多属性比较乱，清洗了例如日期这样的属性方便特征提取；加入了信息完整度特征，很好地利用到了含有空值的样本；对于order这个id含有重复的样本，尝试了提取特征后按时间去重和按第一条和最后一条去重，发现按第一条去重效果是最好的，很好地使用到了order的信息；通过特征的重要性排序筛选了特征，也发现了提取的业务相关的特征是最重要的。

### 7.2 模型

模型的创新点主要体现在模型融合上。考察指标为AUC，侧重于答案的排序。在进行加权融合时，先对每个模型的结果进行了归一化，融合效果很好。

## 八、赛题思考

清洗数据非常重要，像时间这样的属性非常乱，处理起来也比较麻烦，我们只是简单地进行了处理，如果能够更细致的处理效果应该更好；某些属性，例如hobby，内容太复杂没有使用到，但这个属性肯定包含了许多有价值的信息，但遗憾没有发现一个好的处理方式。