“魔镜杯”风控算法大赛模型说明

——“我这么纯洁根本听不懂”

风险算法赛题解读

数据处理方案

变量衍生方案

数据集合划分方案

模型算法

讨论

“魔镜杯”风险算法赛题解读及算法设计

题目：以8万、平均400维度数据样本为训练集合，结合各用户当前的信用状态，给每个借款人打出当前状态的信用分，在此基础上，再结合1万新样本信息，打出对于每个标的6个月内逾期率的预测。

我们小组将题目解读为以下三点：

1. 通过给定的8万样本集合预测1万样本中每条样本的违约概率
2. 以将用户划分为合理的客户群体为目标，进行数据处理、模型选择等工作
3. 针对此问题，我们选择使用树模型为主，线性模型为辅的模型设计方案

数据处理方案（基本清洗）

在对原始数据的检测过程中我们发现以下几个问题：

1. Master数据中包括4各城市变量，其中UserInfo_2和UserInfo_20比较相似，而且与UserInfo_24中的细节地址信息强相关，因此使用UserInfo_2和UserInfo_24来填充UserInfo_20中的缺失数据；同理使用UserInfo_4来填充UserInfo_8的缺失数据。
2. ListingInfo数据中存在数据格式差异，需要采用统一的日期模式进行处理，并转化为时间戳。
3. 数据中包含若干变量，其中缺失值占比超过99%，或者同一属性（不是缺失值）占比超过99%，为了防止小样本对模型的干扰，我们将其清除。
4. 在UserUpdate数据集合中，有一部分属性含义相同，但是因为大小写的原因，可能导致误分类，因此将本部分数据实现转换为小写形式。

数据处理方案（二项分布检验）

清洗数据集合中明显不合理的数据后，我们希望从数据集中提取更加有效的信息。考虑到本问题是一个0/1分布问题，因此可以使用二项分布检验对原始数据集合进行观察，具体方案如下：

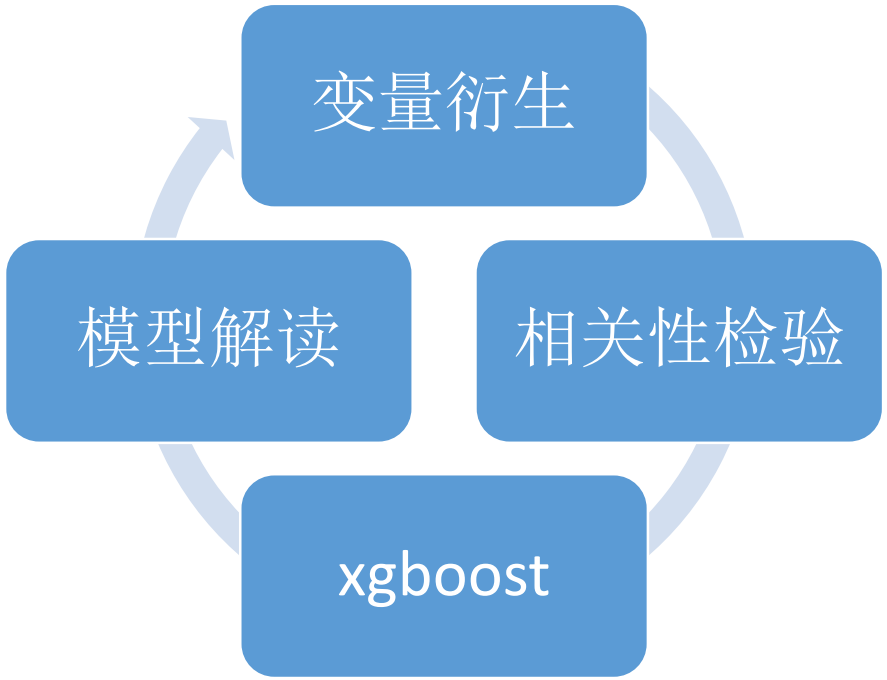
1. 统计全部样本目标的0/1分布情况，作为总体分布p0
2. 统计某个属性对应的样本目标的0/1情况，作为样本分布p1。
3. 使用二项分布检验两个分布的差异性，差异性越强，说明该属性的区分度越高。

在二项分布检验的基础上，我们分别对类别变量和数值变量进行处理：

1. 对于类别变量，我们使用二项分布检验挑选各变量中具有区分度的属性。
2. 对于数值变量，我们使用二项分布检验来判断变量中的空值和异常值。
3. **对于类别变量，我们使用二项分布检验的p值作为新特征，用来描述变量的可信性。**

变量衍生方案（总体方案）

1. 通过xgboost初步建模，观察入选变量，选择重点模块进行特征处理
2. 对于可以人工解读的变量（省份、城市），进行处理。



变量衍生方案（ThirdParty）

我们发现原始数据集中的ThirdParty数据对模型有很重要的影响，我们将其解读为用户的金融消费信息（信用卡信息），通过查找资料我们发现，在信用卡业务的场景中，某些变量的绝对值没有什么明显的意义。比如并不是用信用卡消费多的人就一定比消费少的人更坏或者更好。那些人自有的消费习惯或者家庭条件对他的信用卡消费金额可能影响更大。有的时候一个人在某段时间的消费趋势可能比消费的绝对值更能反映他的风险水平。比如在本次建模数据中我们根据变量名和取值判定third_party相关的变量指的是第三方支付所记录的数据。一共有7大类和十二个月的数据。另外13-17可能是一些汇总数据。我们对1-12月同类的商品消费做趋势分析发现一个特别有趣的规律：如果一个人对某一类商品的消费越来越多他的风险趋势要小于一个消费越来越少的人。因为我们没有接触具体的业务，很难很准确的去解释这种现象。我们的猜测是一个消费越来越少的人要么是自制力不太强，对自己的消费没有理性的规划。要么是财务上出现危机导致无法维持之前的消费水平。这两种情况都会导致风险的增加。求一组数值变量的趋势值比较简单，我们采用的是线性回归。把不同月份的消费作为y，把不同月份转换为数值后作为x进行线性拟合得到系数。

因此我们采用下列方案进行变量衍生：

1. 对于Thirdparty变量，我们发现，1-12和13-17有明显区别，因此对这两部分变量分别计算各自的统计信息。
2. 对于1-12的变量，我们将其解读为不同月份的数据，在此之上，我们将这些数据分割为1-6、7-12、1-3、4-6、7-9、10-12这种不同时段的分组数据，分别计算其统计量（平均值、方差、最小值、最大值、梯度（变动速率））。
3. 计算衍生变量间的相关性，去除明显相关的特征（我们发现平均值与最大值具有很强的相关性）

变量衍生方案（省份/城市信息/LogInfo/UserUpdate）

1. 针对省份信息和城市信息，引入数据补丁，使用各省份的经济数据（gdp以及人均收入等数据）代替原始省份变量。
2. 使用中国城市分级信息，将城市数据转换为城市分级数据。
3. 对于LogInfo和Userupdate数据，我们按照Idx统计每个Idx下各种行为发生的次数，用来反映该Idx的操作行为。

省份	省份编码	gdp	人均GDP	人口	人均收入	房价/收入
北京	11	13	2	26	2	14.5
天津	12	17	1	27	6	8.4
河北	13	6	17	6	22	6.7
山西	14	24	23	19	23	6.1
内蒙	15	15	6	23	10	5
辽宁	21	7	7	14	9	6.3
吉林	22	22	11	21	25	6.2
黑龙江	23	20	18	16	26	7.4
上海	31	12	3	25	1	12.1
江苏	32	2	4	5	4	6.7
浙江	33	4	5	11	3	9.5
安徽	34	14	26	8	14	6.8
福建	35	11	9	18	7	9.1
江西	36	18	25	13	20	7.3
山东	37	3	10	2	8	5.5
河南	41	5	22	1	17	5.6
湖北	42	9	12	9	13	6.9
湖南	43	10	16	7	11	5.5
广东	44	1	8	4	5	8.4
广西	45	19	27	10	15	5.9
海南	46	28	20	28	16	12.3
重庆	50	21	13	20	12	6.8
四川	51	8	24	3	18	7.4
贵州	52	26	29	15	27	5.9
云南	53	23	30	12	21	5.9
西藏	54	31	28	31	30	5.9
陕西	61	16	14	17	19	7.1

变量衍生方案（LogInfo/UserUpdate）

本次建模大赛中除了master数据集外，还有Loginfo和userupdate两个数据集。这两个数据集的特点就是同一客户在不同时间点进行了不同的操作。也就是说每个客户都发生了顺序的状态变化。如果我们假定“坏客户“和“好客户”具有不同的状态变化模式的话，当我们知道一个客户的状态变化后，就可以通过马尔可夫链模型来判断这个人分别是“坏客户“和“好客户”的概率。

具体做法：

1. 先把客户按‘target’分成好坏两组。
2. 把好客户的操作按时间排序构成一个向量，然后取这个向量的每个元素的下一个元素构成一个新的变量。
3. 求这两个向量的混淆矩阵。这个混淆矩阵的元素就是状态变化的概率
4. 同理可以求得坏客户的混淆矩阵。
5. 对一个新的客户的状态变化，可以通过混淆矩阵得到他每次状态变化的概率，然后把这些概率相乘就得到了他整个状态变化的概率（我们只能假定状态变化之间相互独立，不然模型会过于复杂）。比较由两个混淆矩阵得到的概率来判断他更可能属于好客户还是坏客户。

虽然想法很好，但是马尔科夫链在本次建模的表现并不理想。主要原因可能是：

1. 同一天发生了很多操作，这些操作因为没有具体的时间戳，导致我们无法判断它们正确的顺序，而顺序对马尔可夫链至关重要。
2. 这是风控模型而不是防欺诈模型。而这些操作的记录大部分都是在贷款初期。客户的风险表现往往在很后期，所以这些操作与风险的关联就弱了很多。如果是欺诈模型，因为欺诈分子在一开始就有欺诈的意图，如果使用马尔可夫链来进行行为判断，可能会表现得好很多。

变量衍生方案（one hot 与 woe）

根据变量属性数量，我们对类别变量分别采用one hot 和 woe转换。

1. 当变量属性数量比较少时，我们使用one hot转换。
2. 有些离散变量取值太多，直接进行one hot转换可能造成维度灾难。所以我们为了更好利用离散变量的信息，决定直接用离散变量中不同取值下的违约率作为该取值的值。比如地区变量中，我们直接用省份的违约率代替这个省份。

离散变量连续化的优点：

1. 避免维度灾难。
2. 尽可能的利用了离散变量的单变量信息。

离散变量连续化的缺点：

1. 连续化后，变量之间的共线性会增加。
2. 连续化后，无法挖掘到变量之间的交互作用。

离散变量连续化的注意点：

如果某个取值内的样本量太少，则该取值下的违约率不稳定，所以需要设定一个阈值，少于该阈值样本数的取值要和其它取值合并到一起。我们的经验值是30。

数据集合划分方案

考虑到初赛中daily数据和final数据的分布差异性比较大，我们将8万训练数据分为三部分，采用三步验证的方案来检查模型的稳定性。

数据划分方案：

1. Train数据（6万：初赛训练集 + 复赛训练集）
2. Test_A数据（1万：初赛daily数据）
3. Test_B数据（1万：初赛final数据）

三步验证方案：

1. 通过Train数据内部交叉验证，选择参数
2. 预测Test_A数据，并计算AUC值衡量预测结果
3. 预测Test_B数据，并计算AUC值衡量预测结果

由于训练集数据量充足，因此7万的训练结果与6万的训练结果不会有过大的差异。与此同时，考虑到daily数据和final数据分布差异比较大，因此我们选择使用两个样本外验证集合，用来控制模型的稳定性。

模型算法

模型	交叉验证（4-fold cv）	A_test	B_test
Xgboost(dt).1	0.7782	0.7737	0.7780
Xgboost(dt).2	0.779	------	-----
Xgboost(dt).3	0.782	0.7822	0.7835
Xgboost(lr)	0.7516	0.7497	0.7510

本次比赛采用xgboost_GBDT余xgboost_LR相结合的方案，并采用ranking average进行预测结果的集成和组合。

2*xgboost(dt).3 + 1.5 *xgboost(dt).2 + 1 * xgboost(dt).1 + 0.05 * xgboost(lr)

讨论

1. 在处理二分类问题时，二项分布检验可能是一种很有效的方法，其p值既可以反映属性的信息量也可以反映属性的稳定性与可信性，因此可以考虑对二项分布检验进行深入研究；
2. 由于我们小组将本类问题看做客群划分问题，因此给予了决策树模型更高的投票权重，但是由于决策树模型调参复杂，而且反馈速度不如线性模型，因此这可能使我们小组接下来要改进的重点；
3. 由于对数值变量处理还比较初步，缺乏类似二项分布检验的评价指标，我们对数值变量信息提取还有工作要做。