广告位招租

西南财经大學

西南财经大学“新网银行杯”数据科学竞赛暨第十七届统计建模大赛

报告书

队长：刘剑刚

队员：鲁宇星、唐海、李云鹏

2020 年 11 月 9 日

摘要

四川新网银行以建设“数字普惠银行”为愿景，运用互联网大数据风控、云计算、人工智能等技术，为客户提供具有高可得性和良好用户体验的金融产品。综合利用客户的信用数据、行为数据等信息建立高风险客户识别模型可帮助金融机构及时发现风险并减少损失，因此，如何精准识别高风险客户是金融机构风险管理关注的重要问题。

本次竞赛我们主要采用 LightGBM 建模，经过数据分析，数据清理，以及反复进行特征工程，不断找寻有利于模型预测的特征，在竞赛指标使用变形的 F1_Score 下，线上获得成绩 0.4138，最终榜上排名 6/299，最终小组初赛排名为第 4 名。

关键词：高风险 精准识别 LightGBM

目录

1 赛题分析

1.1 赛题背景

四川新网银行以建设“数字普惠银行”为愿景，运用互联网大数据风控、云计算、人工智能等技术，为客户提供具有高可得性和良好用户体验的金融产品。综合利用客户的信用数据、行为数据等信息建立高风险客户识别模型可帮助金融机构及时发现风险并减少损失，因此，如何精准识别高风险客户是金融机构风险管理关注的重要问题。

1.2 数据解析

在本次比赛中，比赛官方给出的数据包括两种，分别是客户的基本信息和截至某个时间观察点的客户行为信息，且这些数据都是真实业务场景下的脱敏数据，包含多产品（客群）的高维特征数据和面板数据（部分截面数据，部分面板数据）。在比赛中，需要参赛者基于客户的基本信息和行为数据信息运用统计、机器学习算法等工具建立模型，识别高风险客户和低风险用户，因此这是一个二分类任务。首先，观察数据的基本情况：

表 1-1 用户标签 0 和 1 的数量

target	count
0	13657
1	1623

从表 1-1 可知，标签文件包含有两个字段，分别是 id 和 target，而 target 就是我们需要预测的目标，其中 0 为低风险客户，1 为高风险客户。可以看到，标签的分布极不均衡。

用户基本信息表（下面简称 B 表），由于特征数较多，我们只显示了后续保留的特征。

表 1-2 B 表特征基本统计信息

	x_num_0	x_num_1	x_num_2	x_cat_0	x_cat_3
mean	0.0417	-1.0332	-3.4493	0.2094	0.1679
std	0.0616	12.9655	19.9633	0.4069	0.3738
min	0	-99	-99	0	0
50%	0.0208	0.674	0.769	0	0
max	1	1	1	1	1

从上表中可以发现，B 表的这些数值型特征（以 x_num 为前缀）中，特征 x_num_1、x_num_2 和 x_num_3 出现了-99 的异常值，而其他数值型特征的值都是在[0,1]之间，说明这些数据都是已经进行了归一化处理,而-99 是代表了缺失值。B 表的这些类别型变量（以 x_cat 为前缀）中，取值都是只有 0 和 1，用户的行为信息表（下面简称 M 表）中的情况也如上述 b 表中的分析一样。

另外，通过观察 M 表的数据发现，同一个用户下面存在着有重复的时间点，即相同 id 下的 timestamp 特征中，存在着有两个或多个从 0 开始计数的时间段，统计了具有重复的时间点的用户数量在训练集和测试集中的占比，具体情况如图 1-1 所示。

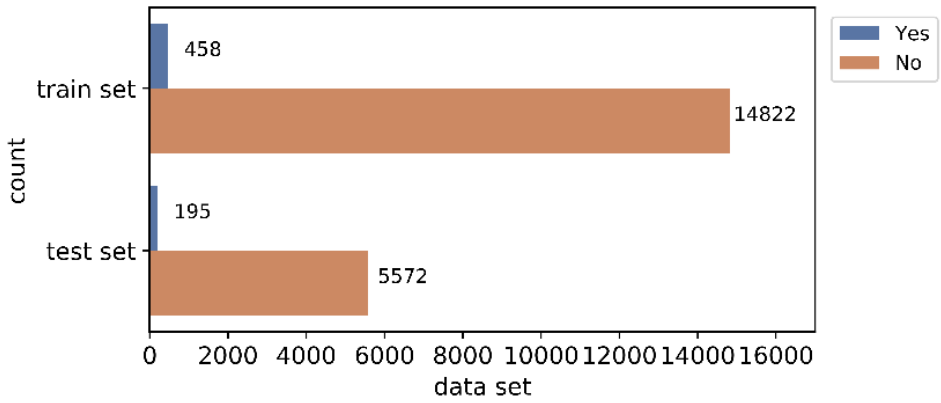

图 1-1 冗余 timestamp 统计

从上面这幅有无重复时间点的样本数量之间的对比图中可以看出，含有重复时间点的用户只占总用户量的极小一部分，大约只有 0.03。考虑这也许是脏数据，后面具体实施中，可以考虑将其删除。对于 M 表还分析了各个特征有多少个不同的值（这里为了方便进行分析，采用的是各个用户据当前最近一个时间点的数据）。图 1-2 展示了表 1-3 中特征之间的关系。

表 1-3 M 表中 unique 的数量大于 10000 的特征

column	unique
x_num_13	13580
x_num_18	13293
x_num_20	12751
x_num_30	12151
x_num_43	13293
x_num_46	12151
x_num_47	14698

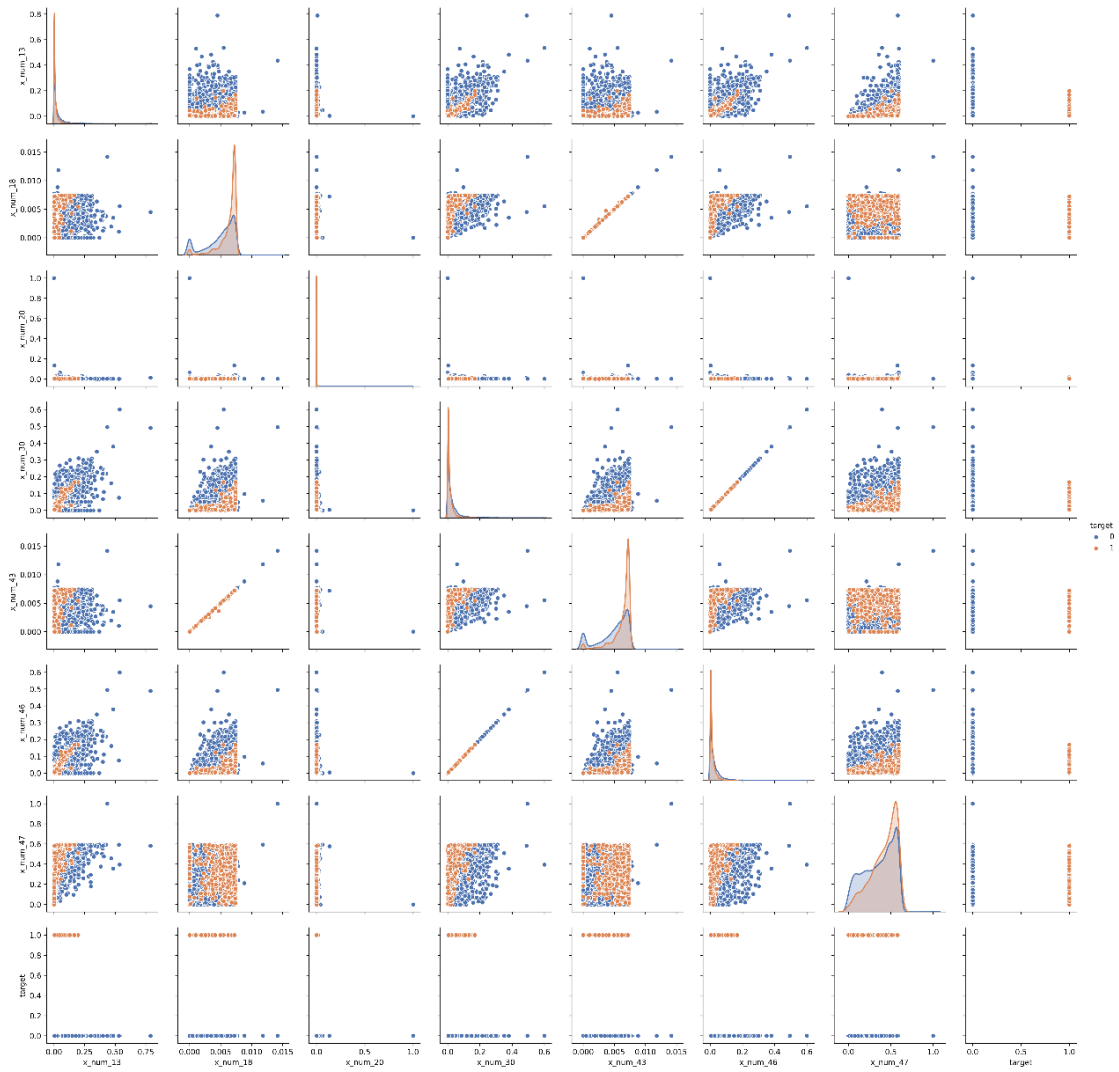

图 1-2 特征关系图

从上面的关系图中可以看出，x_num_30 和 x_num_46，x_num_18 和 x_num_43 这几个特征之间是高度正相关的，而其他特征之间的相关性都不高，后面可以考虑去掉冗余度非常高的特征（实际未进行删除，由于后面采用 LightGBM 模型，在共线性特征数不多的情况下，由于集成模型的特点，可以规避掉此影响）。

查看各个特征在训练集和测试集上的分布，如图 1-3 所示。

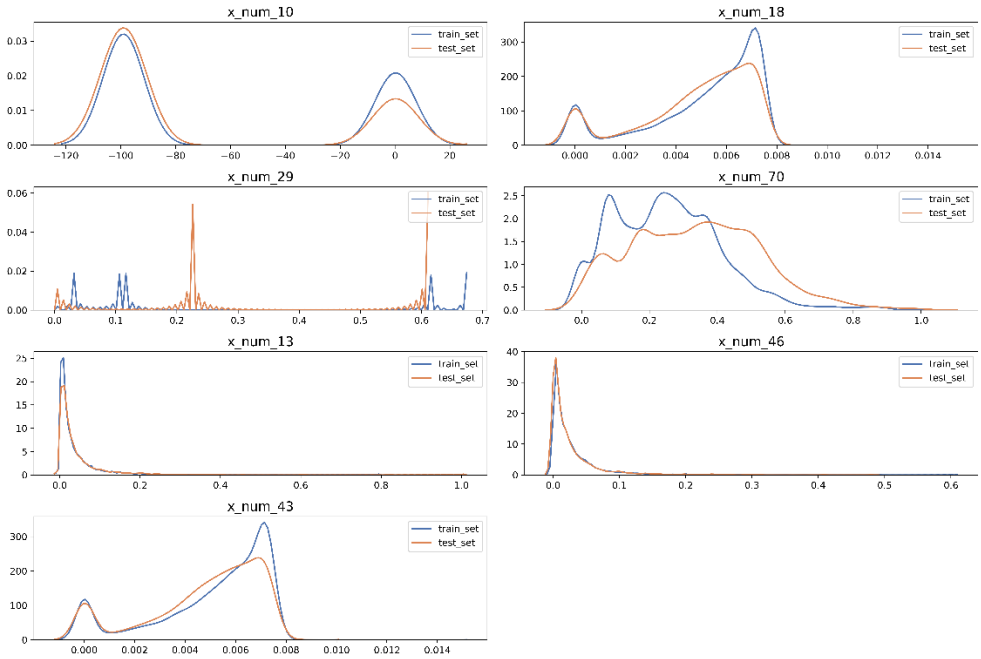

图 1-3 训练集和测试集特征分布

从上面的几幅图像可以看出，训练集与测试集中特征的值的分布存在着差异，初步判断测试集中标签 0 和 1 之间的占比也会有所不同。

为了验证这个猜测，我们使用了对抗验证的方法。一般用来评估模型效果的方法是交叉验证，但是当样本分布发生变化时，交叉验证就无法准确评估模型在测试集上的效果，会导致模型在测试集上的效果远低于训练集。而对抗验证方法，就是主要试图来寻找和测试集相似的样本，若能找到与测试集相似的样本，则在线下就可以将训练集按照前述经验进行划分，以保证线下、线上验证的相似性。该方法的具体步骤如下：

① 将测试集与训练集组合到一起，形成一个新的数据集；

② 新增一个标签列，将测试集数据的标签置为 1，训练集的数据置为 0；

③ 构造一个分类器，例如逻辑回归，决策树或 LightGBM；

④ 观察模型的效果，如果模型的 AUC 超过了 0.7，则说明训练集和测试集的分布存在着较大的差异（AUC 为 0.5 为训练集和测试集分布较一致的结果）；

⑤ 然后将训练集数据中的预测样本概率最接近 1 的样本作为验证集，其他数据继续作为训练集。

在这次比赛中，我们按照上述的步骤进行了验证，将数据送入 LightGBM 分类器进行训练，最后 AUC 得分是 98%，说明训练集与测试集之间数据的分

布存在着较大的差异。此外，按照此方法，我们可以选择预测概率值较高的训练集来作为验证集。然而遗憾的是，我们经过测试，发现只有几百个的训练集中的样本，预测概率值较高，但这样的大小来说，验证我们模型的准确性，是远远不够。这也极大难度的增加了我们验证线下新构造特征是否有效的难度。

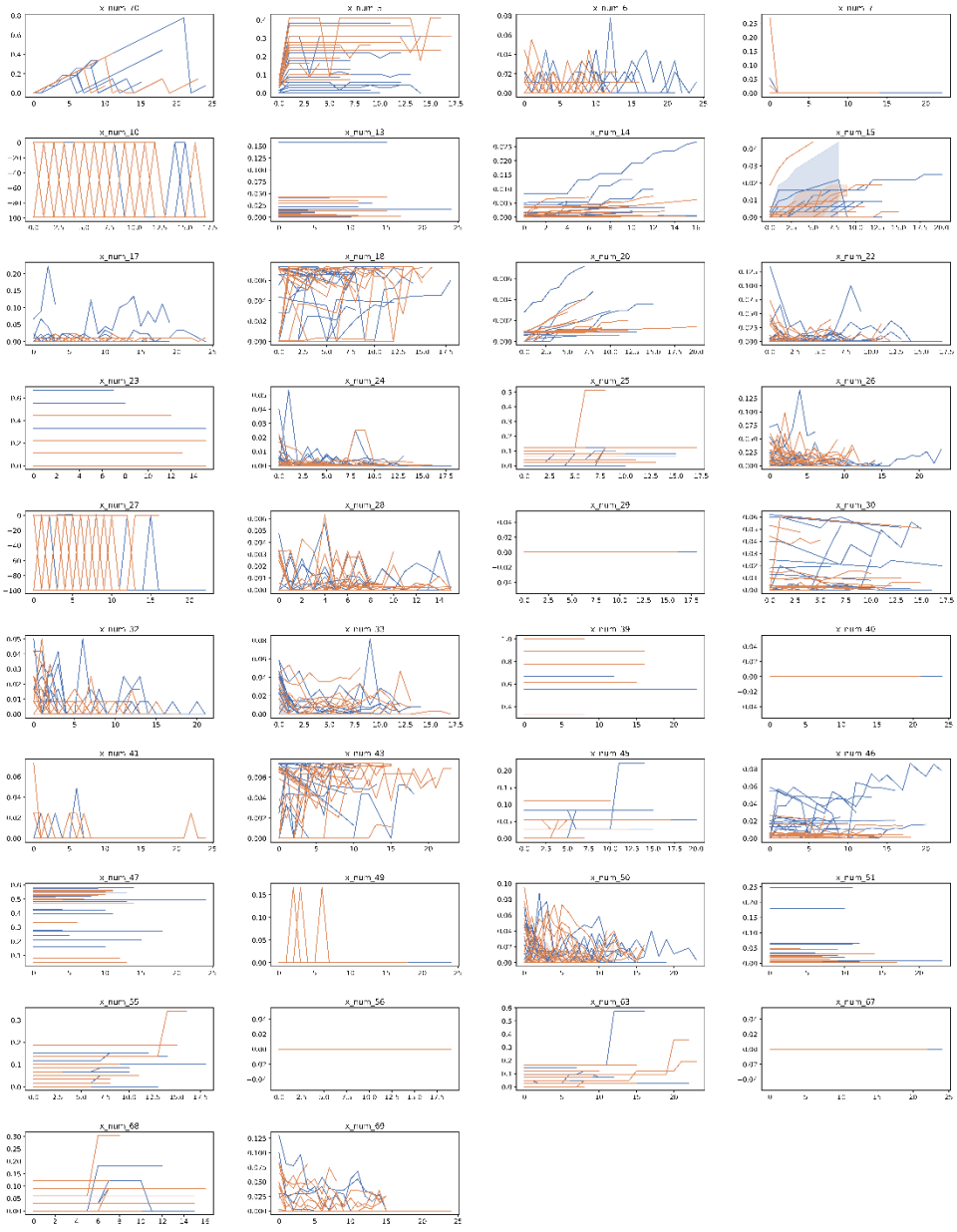

图 1-4 M 表特征样本内变化趋势

如图 1-4 所示，其中绿色的线代表 0 标签样本，橘红色的线代表 1 标签样本。我们对每个标签，抽取 15 个样本（主要考虑到版面美观问题，实际观察时，我们采用 0、1 标签各 300 个样本来观察变化趋势）来观察不同类别下的不同样本的每个特征的变化规律。可以显而易见看到的是，M 表当中的其实并无

明显有区分度的特征，不同标签的样本特征变化交错在一起，这大概也解释了，为什么在后续我们试图构造关于时间序列相关的特征，对整体模型预测，并无很大效果的原因。为此，在后续对 M 表特征处理的时，我们主要考虑使用的是统计特征。

此外，对赛方提供的 score 得分函数进行分析，其中 precision 是准确率，recall 是召回率。得分函数：

$$\text{score} = \frac{(precision * recall)}{(0.4 * precision + 0.6 * recall)}$$

其中，

$$precision = \frac{tp}{tp + fp} \quad reacll = \frac{tp}{tp + fn}$$

代入 score 得分函数可得，

$$\text{score} = \frac{tp^2}{tp^2 + 0.4 * tp * fn + 0.6 * tp * fp}$$

$$= \frac{1}{1 + \frac{0.4 * fn + 0.6 * fp}{tp}}$$

根据预测结果，我们又可以分为 4 种情况：

① 0 → 1(正预测为负) → $\begin{cases} fn + 1 \\ tp - 1 \end{cases}$ 根据公式可知 score 得分将降低

② 0 → 0(正预测为正) → $\begin{cases} fn - 1 \\ tp + 1 \end{cases}$ 根据公式可知 score 得分将提升

③ 1 → 0(负预测为正) → $fp + 1$ 根据公式可知 score 得分将降低

④ 1 → 1(负预测为负) → $fp - 1$ 根据公式可知 score 得分将提升

由于权重的影响，正确预测对 1 将会比正确预测对 0 的提升更为明显。

2 数据预处理

2.1 缺失值分析

查看数据集，发现-99 为缺失值标志，将-99 替换为 nan 值。连接 M 表与 B 表，对每个特征查看缺失率，见表 2-1。

表 2-1 训练集和测试集缺失率

特征名	训练集缺失率	预测集缺失率
x_num_19	100.00	100.00
x_num_38	100.00	100.00
x_num_10	62.331600	63.331340
x_num_27	62.331600	63.331340
x_num_2	5.015598	4.866122
x_num_3	1.779896	1.630456
x_num_1	1.779896	1.630456

由表 2-1 可以看出，在训练集与测试集中，特征 x_num_19 和 x_num_38 的缺失率达到 100%，可能属于噪声。而对于缺失率达到 60%的特征 x_num_27 和 x_num_10 则需要继续进行分析。

将所有特征放入基础的 LightGBM 模型中进行训练，依据树模型的叶子分裂得到重要性，查看 7 个缺失特征的重要性。

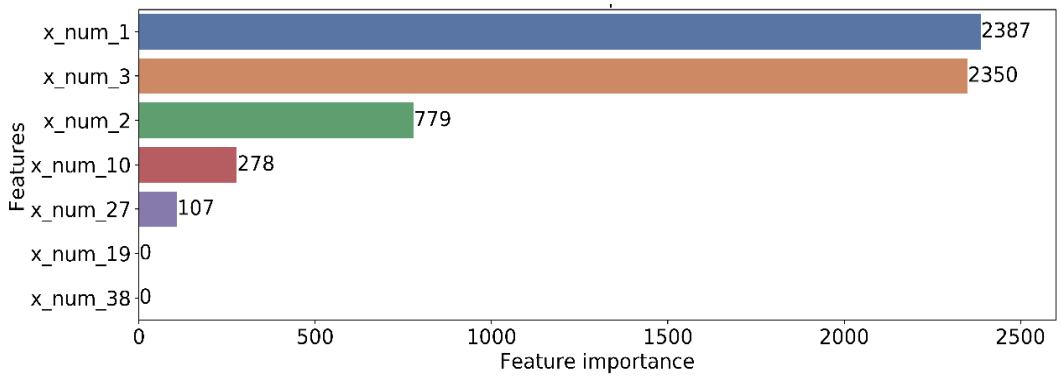

图 2-1 存在缺失值的特征的重要性

由图 2-1，可以看出特征 x_num_19 和 x_num_38 的重要性为 0，而其余特征对模型产生了影响，所以确定这两个特征为无用特征，进行丢弃。

2.2 缺失值处理

对于缺失值通常会使用平均值、中位数、众数进行填充。尝试对其余缺失特征进行填充，以缺失较多的 x_num_27 和缺失较少的 x_num_2 为例，展示在训练集上，将-99 替换为 nan 值、平均值和中位数后的密度图。

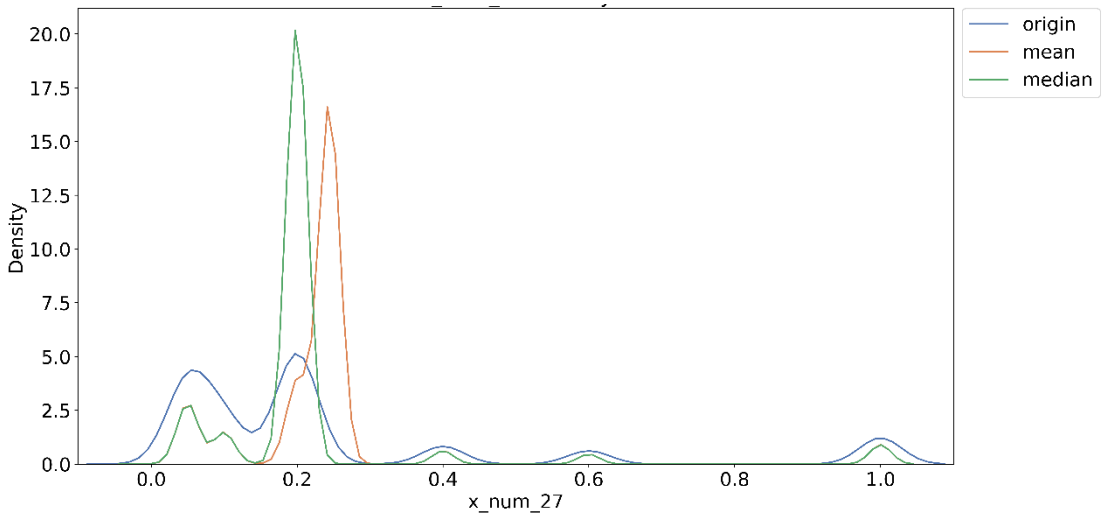

图 2-2 x_num_27 不同填充下的密度图

由图 2-2 可以看出，对于缺失率较多的 x_num_27 类特征，平均值和中位数填充明显影响了原始数据集的分布，所以对于这两个缺失较多的特征，不进行填充。

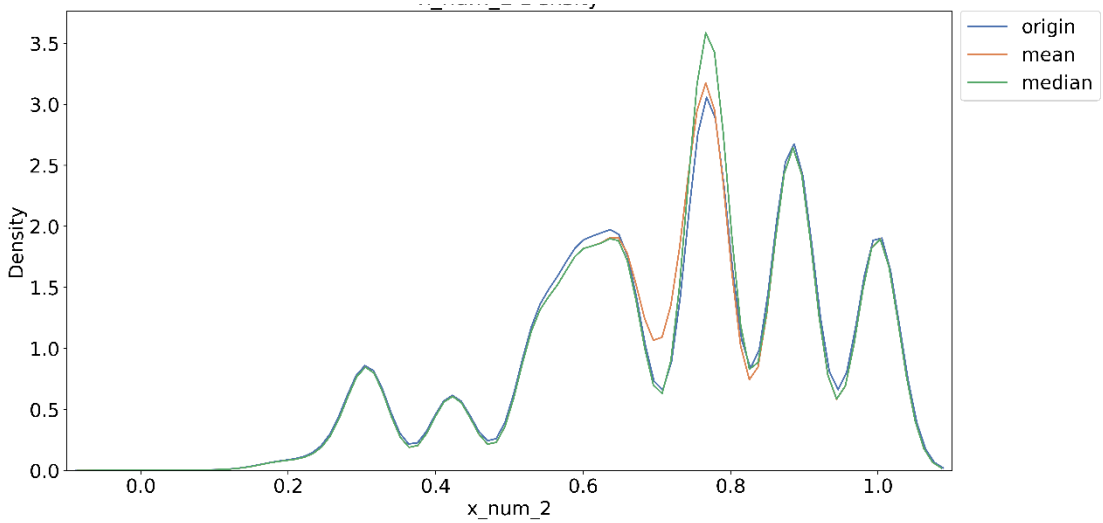

图 2-3 x_num_2 不同填充下的密度图

而对于缺失值较少的 x_num_2 类特征，填充后对于密度分布影响不是很大。然而，填充后的预测结果比起未替换-99 的结果产生了大幅下降。

图 2-3 中说明了在-99 中含有较多的 1，说明含有-99 的特征具有重要影响。另外，结合特征重要性，我们猜测-99 对于特征的表现有很强的作用，所以对训练集的-99 进行保留。

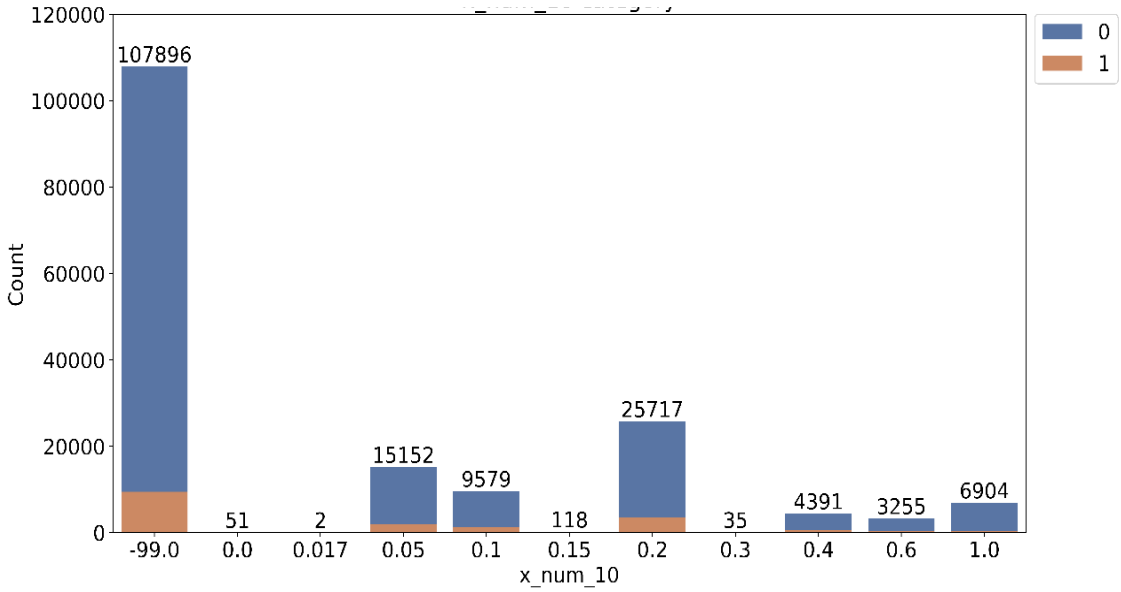

图 2-4 x_num_10 填充

将测试集中的-99 替换成 nan 值后，我们发现预测效果得到了提升。可能是预测集中的-99 并不像训练集的-99 表现效果强，而是属于噪声，所以我们对预测集中的-99 做了同训练集中不一样的处理。

3 特征工程

3.1 基本特征筛选

因为 M 表中每个 id 对应的行为数据的数据不一致，所以我们选取 M 表中时间戳最大的一行行为数据和 B 表的基本信息数据合并后作为初始特征。然后再进行特征筛选。

经过对数据的分析，发现数据内存在一些同值的数据。所以首先需要对数据进行同值化处理，对 B 表和 M 表进行分析后，我们删除了 B 表中 x_cat_5，和 M 表中 x_num_19、x_num_38。

为了进一步筛选出有用的特征，我们使用最大互信息系数（MIC）、皮尔森相关系数（Person）、距离相关系数（Distance）来计算计算剩余特征的和 y 标签的相关性。三种方法的对比如下表 3-1 所示。

表 3-1 算法对比

算法	用途	适用范围
最大互信息系数	衡量两个变量之间的关联程度	线性、非线性数据
皮尔森相关系数	衡量两个变量之间的线性相关程度	线性数据
距离相关系数	为了克服 Person 相关系数弱点	线性、非线性数据

表 3-2 B 表计算结果

columns	MIC	Person	Distance	columns	MIC	Person	Distance
x_num_0	0.68	0.38	0.7	x_cat_2	0.01	0.02	0.14
x_num_1	0.54	0.01	0.11	x_cat_3	0.01	0.01	0.12
x_num_2	0.34	0.03	0.18	x_cat_4	0.04	0.09	0.28
x_num_3	0.28	0.01	0.12	x_cat_9	0.02	0.04	0.19
x_cat_0	0.07	0.13	0.33	x_cat_10	0.04	0.07	0.26
x_cat_1	0.01	0.01	0.1	x_cat_11	0.09	0.16	0.38

表 3-3 M 表计算结果

columns	MIC	Person	Distance	columns	MIC	Person	Distance
timestamp	0.04	0.03	0.19	x_num_30	0.92	0.2	0.48
x_num_70	0.15	0.15	0.41	x_num_31	0.01	0.03	0.18
x_num_5	0.65	1	1	x_num_32	0.12	0.03	0.37
x_num_6	0.13	0.03	0.43	x_num_33	0.09	0.03	0.24
x_num_7	0.01	0.01	0.1	x_num_39	0.36	0.03	0.18
x_num_10	0.15	0.24	0.46	x_num_40	0.01	0.03	0.18
x_num_13	0.87	0.26	0.54	x_num_41	0.02	0.03	0.21
x_num_14	0.78	0.11	0.53	x_num_43	1	0.61	0.76
x_num_15	0.09	0	0.21	x_num_45	0.04	0.07	0.23
x_num_17	0.14	0.03	0.44	x_num_46	0.92	0.2	0.48
x_num_18	1	0.61	0.76	x_num_47	0.72	0.21	0.42
x_num_20	0.78	0	0.3	x_num_49	0.01	0.03	0.18
x_num_22	0.05	0.01	0.15	x_num_50	0.06	0.03	0.21
x_num_23	0.38	0.67	0.75	x_num_51	0.73	0.39	0.7
x_num_24	0.69	0.05	0.32	x_num_55	0.08	0.09	0.28
x_num_25	0.07	0.09	0.27	x_num_56	0.01	0.03	0.19
x_num_26	0.09	0.03	0.25	x_num_63	0.09	0.13	0.34
x_num_27	0.14	0.24	0.46	x_num_67	0.01	0.02	0.17
x_num_28	0.66	0.01	0.28	x_num_68	0.07	0.11	0.31
x_num_29	0.01	0.03	0.17	x_num_69	0.07	0	0.12

三种方法的计算结果如表 3-2、3-3 所示，由于特征较多，表中只展示了部分计算结果。

根据每个特征和 y 标签的 MIC、Person、Distance 三个计算结果中，只要有一个值大于 0.1 就保留该特征。初步特征选择后，然后根据 LightGBM 的 feature_importance()函数，可以计算出特征的重要性。在特征重要性的计算结果中，x_cat_1, x_cat_2, x_num_29, x_num_31, x_num_40, x_num_56, x_num67 的值为 0，说明对模型构建没起到作用。因此我们可以进一步筛选出有用的特征。进一步特征筛选后，B 表和 M 表各剩余以下特征表 3-4。b_col 代表 B 表剩下的特征，m_col 表示 M 表剩下的特征。

表 3-4 特征筛选结果

table	columns
b_col	id,x_num_0,x_num_1,x_num_2,x_num_3,x_cat_0,x_cat_3,x_cat_4,x_cat_9, x_cat_10,x_cat_11
m_col	id,timestamp,x_num_70,x_num_5, x_num_6,x_num_7,x_num_10,x_num_13, x_num_14,x_num_15,x_num_17,x_num_18,x_num_20,x_num_22,x_num_23, x_num_24,x_num_25,x_num_26,x_num_27,x_num_28,x_num_30, x_num_32,x_num_33,x_num_39, x_num_41,x_num_43, x_num_45,x_num_46,x_num_47,x_num_49,x_num_50,x_num_51,x_num_55, x_num_63, x_num_68,x_num_69

3.2 聚合特征构造

查看 M 表中不同时间下不同特征 0 的数量，见图 3-1，发现 0 的数量大致相近。实际计算，所有 ID 第一天特征中 0 的总数为 552044，最后一天 0 的总数为 586439，但考虑到最后一天在违约的情况下可能包含更多信息，而且实际提交结果显示最后一天效果表现更好，所以加入每个 ID 最后一天的行为特征。

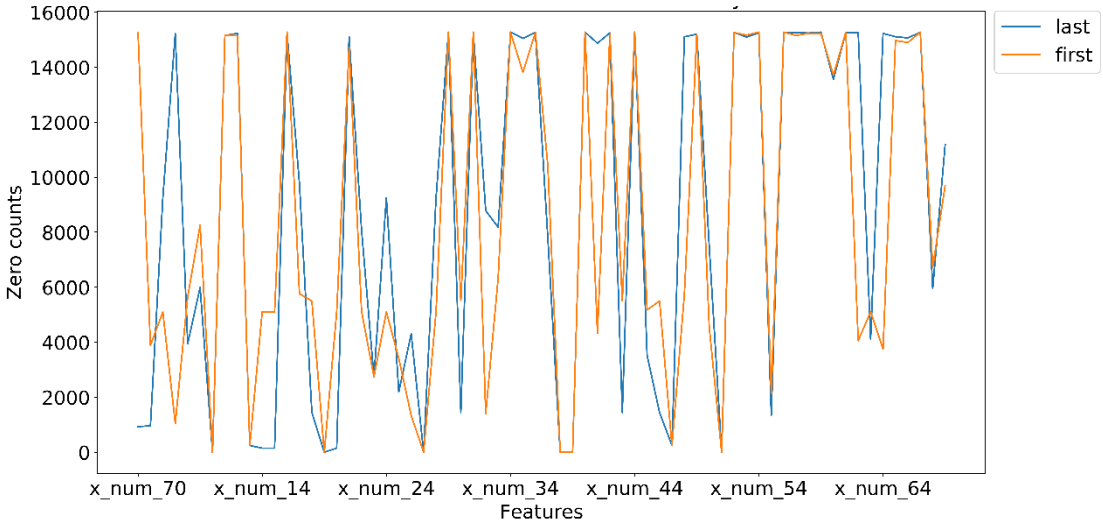

图 3-1 不同 Timestamp 下特征中含 0 的数量

M 表中包含大量行为信息，通常为了获得行为信息的衍生特征，会计算同一 ID 在窗口期内的平均值，中位数，最小值，最大值，标准差和离散系数作为特征。在此训练集上，最小值包含大量的 0 被我们剔除；平均值在实际中表现较差被剔除；中位数，最大值，标准差和离散系数表现效果良好被保留。特征的具体描述见表 3-4。特征在 LightGBM 模型下的 gain 值和 split 值，见图 3-2 和图 3-3。

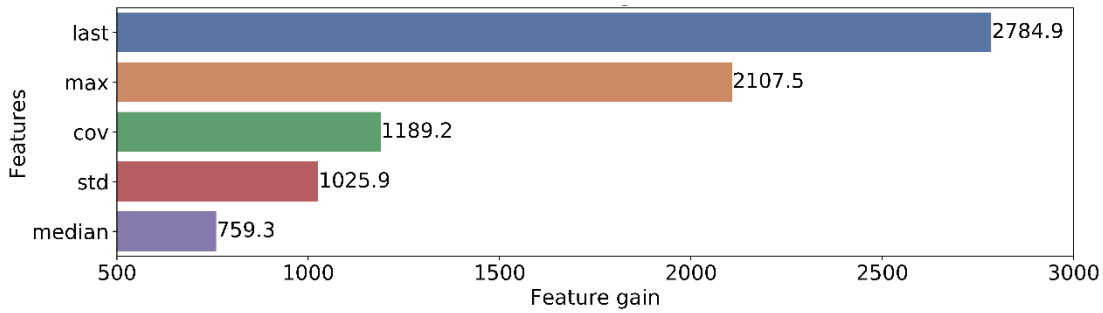

图 3-2 聚合特征 Gain 值

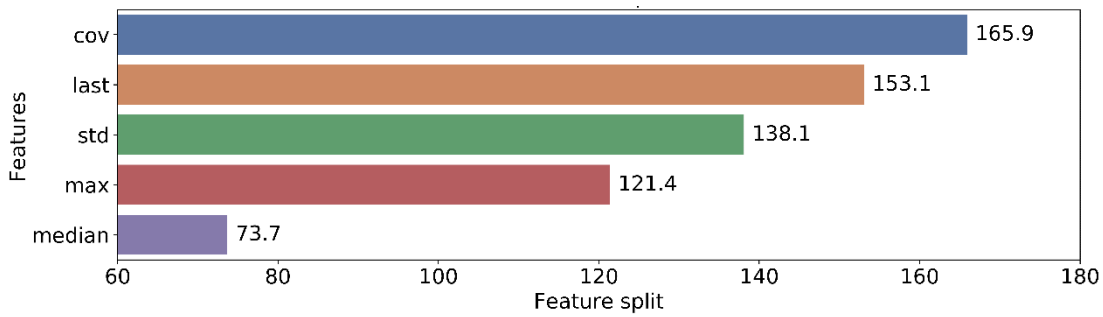

图 3-3 局和特征 Split 值

表 3-4 聚合特征构造表

特征名	描述
Last	选取 m 表中同一 ID 下每个特征在窗口期内最近时间点的值
Max	选取 m 表中同一 ID 下每个特征在窗口期内的最大值
Median	选取 m 表中同一 ID 下每个特征在窗口期内的中位数
Std	选取 m 表中同一 ID 下每个特征在窗口期内的标准差
Cov	选取 m 表中同一 ID 下每个特征在窗口期内的离散系数

3.3 转换特征构造

由于是匿名特征，我们无法得知特征与特征之间的关联，只能通过暴力构造，来进行筛选。在每次组合特征时，我们一次通过构造同一种类型的特征，添加进已有的最优模型来进行验证。通过这种逐步式特征添加的策略，虽然说无法保证所有特征组合为最优组合，但是提高搜索效率，也尽可能让我们快速发现有益于模型的特征。我们最终所确定的添加的转换特征如表 3-5 所示。

表 3-5 转换特征表

特征	解释
mean / last_day	按 id 分组求均值 / 每个 id 的最大 timestamp 的值
max + last_day	按 id 分组求最大值/每个 id 的最大 timestamp 的值
last_day*median_group_by_cat	每个 id 的最大 timestamp 的值*按 x_cat_0 分组求得的中位数
last_day - median_group_by_cat	每个 id 的最大 timestamp 的值-按 x_cat_9 分组求得的中位数
x_cat_XOR	x_cat_0, x_cat_10, x_cat_11 两两异或
x_num_0 * x_cat_9	特征四则运算
x_num_47_last * x_cat_0	特征四则运算
num / sum	含有不同值个数超过 10000 的特征作百分比运算
log	含有不同值个数超过 10000 的特征作 log 运算

我们使用 LightGBM 建模，LightGBM 是一种基于 GBDT 的性能非常好的机器学习模型框架，我们对于特征是否有益于模型，可以通过查看 Gain 增益和 Split 次数，来做评估。其中 Gain 增益表示特征在它所有分裂使用中带来的增益和，Split 次数表示特征在模型中被使用的次数。具体转换特征所带来的的 Gain 增益和 Split 次数，如图 3-4,3-5 所示。需要注意的是，我们所采用的的方

法是逐步式特征添加策略，所以所展示的特征获得 Gain 增益和 Split 次数，是按照添加顺序下模型给出的，并非是最终模型所给出的 Gain 和 Split 次数。

通过图可以看到 num/sum 这个特征无论是在 Gain 还是 Split 都在模型决策中提供较多的支持，但我们线上提交的结果却是在 x_cat_XOR 和 log 特征时，得分有较大的提升。

由于本数据集的不平衡性，我们猜想这应该是由于后两个特征对预测标签为 1 的样本有较大的帮助，而 num/sum 特征有益于标签为 0 的样本，由于得分函数，对正确预测为 1 时给予更大的权重，所以会产生这样的结果。

因此，单单看模型所给出的特征重要性来筛选特征是否重要，不是一种非常妥帖的处理办法。关于我们在线下，如何进一步评估特征有利于模型，并进行线上提交，我们将在第 4 章讨论。

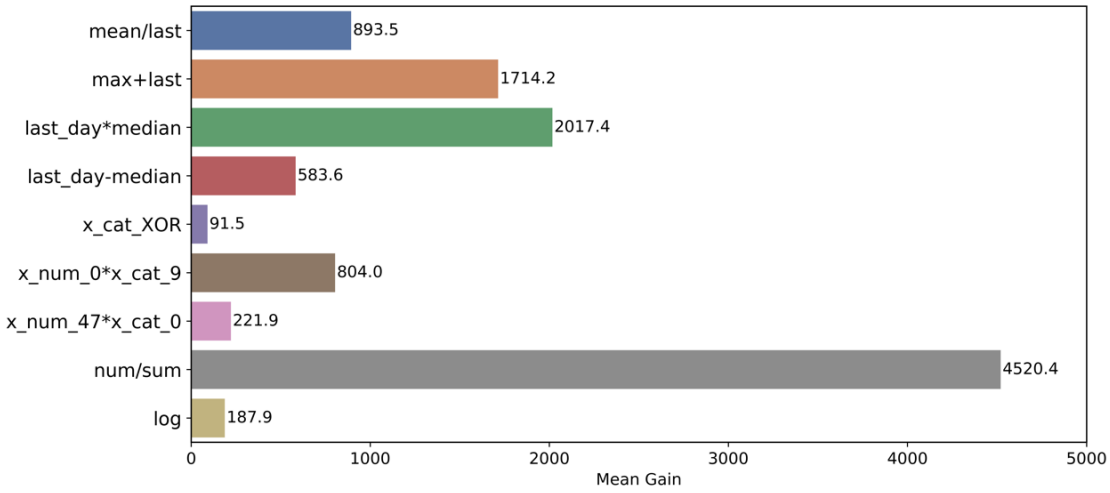

图 3-4 转换特征平均 Gain 增益

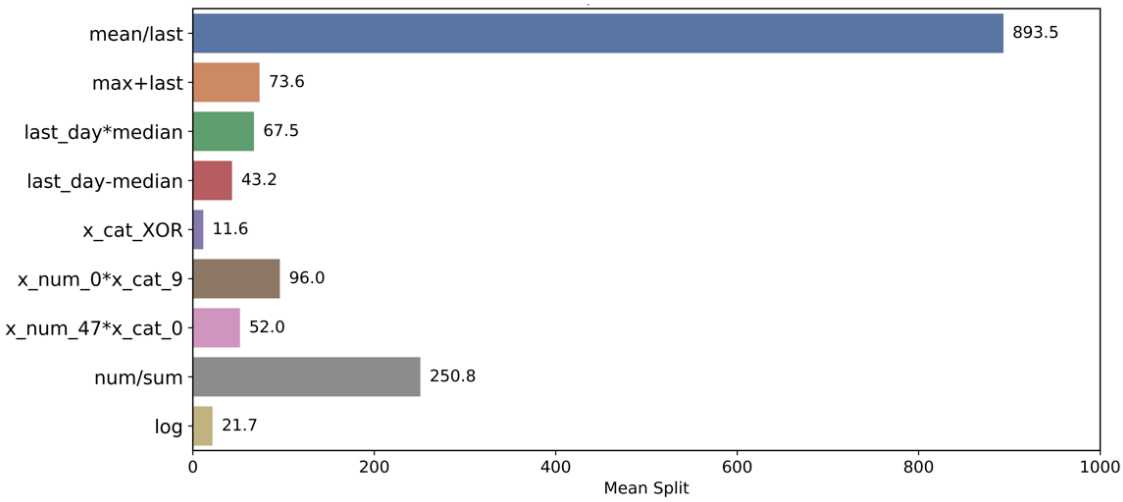

图 3-5 转换特征 Split 次数

3.4 小结

由图 3-6、图 3-7 我们可以看到，特征工程下构造的特征在这个模型中的平均 Gain 增益和平均 Split 次数。

此外，我们也尝试过其他的特征构造，例如利用 LSTM 或者 TCN 来抽取 M 表的信息，希望能将 M 表尽可能多的利用到，结果是利用隐藏层输出结果，十分容易过拟合，这对于这种训练集和测试集分布不一致的预测来说是致命的，因此对于深度学习在此数据集上的运用我们并未探索过多。关于其他特征构造，由于效果并未对模型预测有所提升，就不一一赘述了。

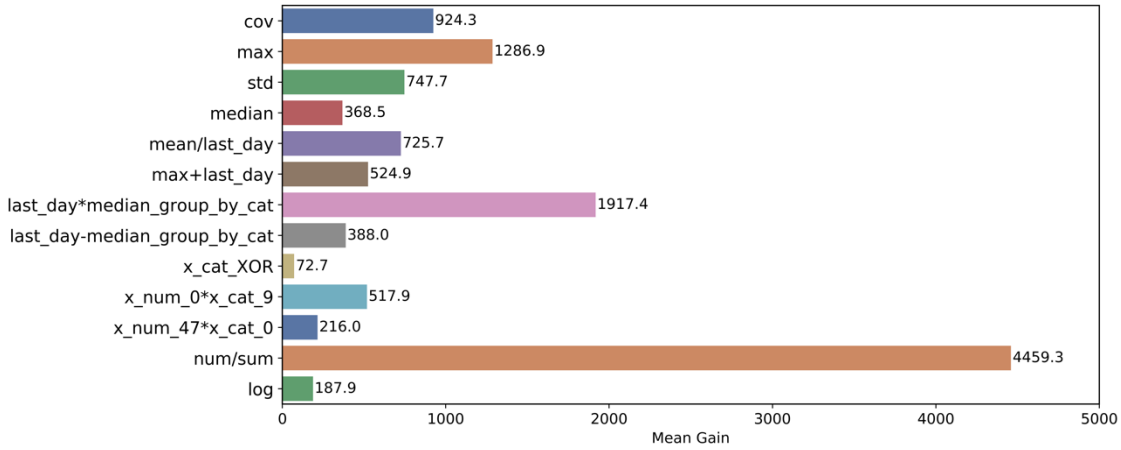

图 3-6 最终模型下特征构造的平均 Gain

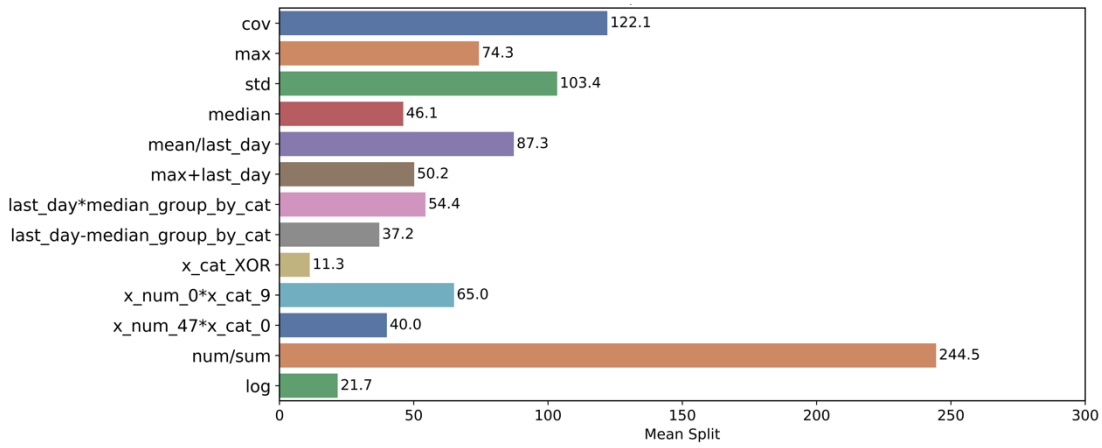

图 3-7 特征构造的平均 Split

最后，需要指出的是，我们在完成整体模型后，进行了特征筛选，利用 LightGBM 的特征重要性，删除 3 个模型（具体见后文）中，Gain/Split 重要性都为 0 的特征，将特征数量进一步删减，最终模型使用特征数为 288。至于，为何在特征工程构建的过程中，没有进行 TopK 的特征筛选，原因一是每次尝试构建

特征数目并不多，二是我们发现在前模型删去一些特征后，会对我们后面新添加特征的结果产生影响，所以，在特征数目，并不算多的情况下，我们在最终确定模型特征后，才进行特征进一步筛选。

图 3-8 为每次添加特征的线上得分情况。

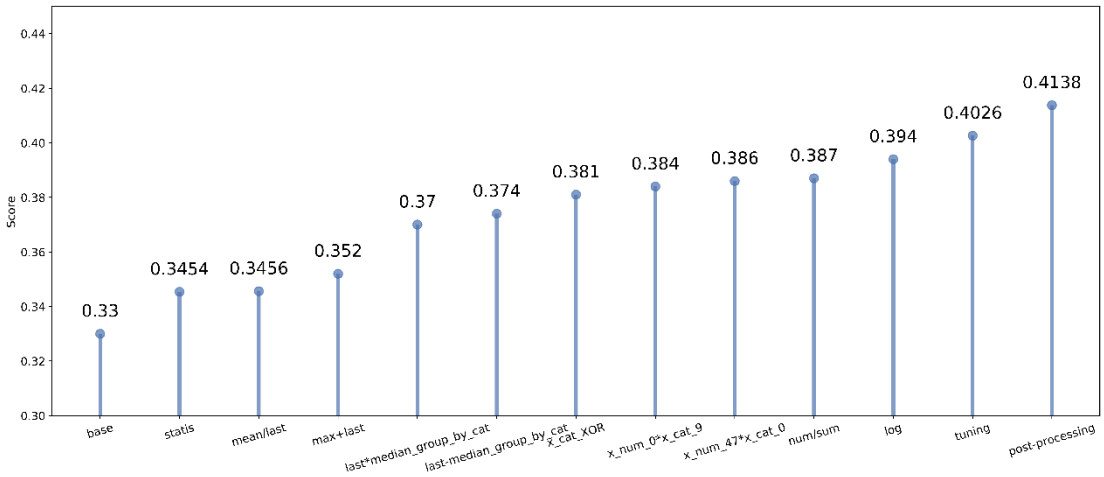

图 3-8 线上的分变化图

4 模型训练

4.1 模型验证的思路及思考

因为每天线上的提交机会有限，不可能尝试所有的想法。在本次比赛中，我们有训练集和待预测的数据集，如果把所以的训练集放进模型训练，我们就无法判断模型的性能，可能会浪费提交机会。所以为了验证我们的思路是否可行，本次比赛初期我们使用 5 折交叉验证的方法，图 4-1 为 5 折交叉验证的示意图。

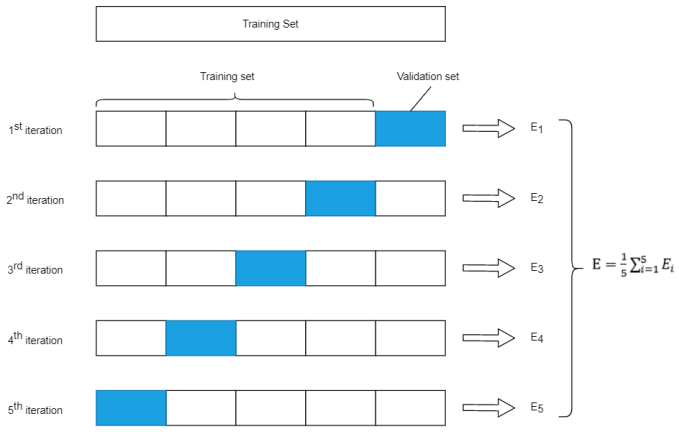

图 4-1 5 折交叉验证

采用 5 折交叉验证，我们需要将原始数据分成 5 组，每次训练模型时，选取其中 4 个 fold 作为 training set，剩下 1 个 fold 作为 validation set，并且每折训练都采用早停（early-stoping）,使用训练好的最佳模型进 test set 的预测，最后对这 5 个不同数据分布的训练子集的预测结果，直接采用结果平均融合。使用五折交叉验证，相对于最开始简单的随机划分验证集有很大的性能提升，并且能够保证模型的稳定性。

我们起初是采用本地交叉验证，并结合逐步特征排除法，逐步将我们构造出来的特征加到模型里面，看构造出来的特征是否对模型有提升。但由于线下线上数据集分布不一致，交叉验证所能保证的是训练集和测试集分布相同时，模型线上线下的稳定。对于本次比赛的数据集来说，交叉验证并未在实际构造中产生很大的作用。

在进行暴力构造特征时，经过多次线上提交后发现，我们发现当加入的特征对模型有提升时，测试数据预测出 1 的个数都有所下降。所以我们开始结合 1 的个数来判断特征是否有效，同时结合最优模型的结果，就是当前最好的提交结果，和它的预测结果进行比较，假如预测出来的结果在 60-70 个不同，我们就把预测结果在线上提交测试。按照上述方法，我们的得分从 0.355 一直提升到了 0.394 左右。

但这种方法，明显不足的是比较依赖线上的反馈结果，但当时苦于没有其他的好办法，便就一直使用此方法进行到比赛结束。

4.2 阈值调整

我们预测利用 LightGBM 构建的分类器，采用输出概率的形式，设置超参 threshold 为 0.25。即

$$y = \begin{cases} 0, & p < threshold \\ 1, & p \geq threshold \end{cases}$$

在初期探索阶段，由于模型训练时候没有添加“is_unbalance”参数设置，所以运行后预测概率更加偏向于 0。为了寻找相对合适的阈值，于是我们使用模型对训练集进行预测，观察了在训练集上，各段预测概率区间的真实值密度图（以 0.05 概率为一个统计区间），如图 4-2 所示。

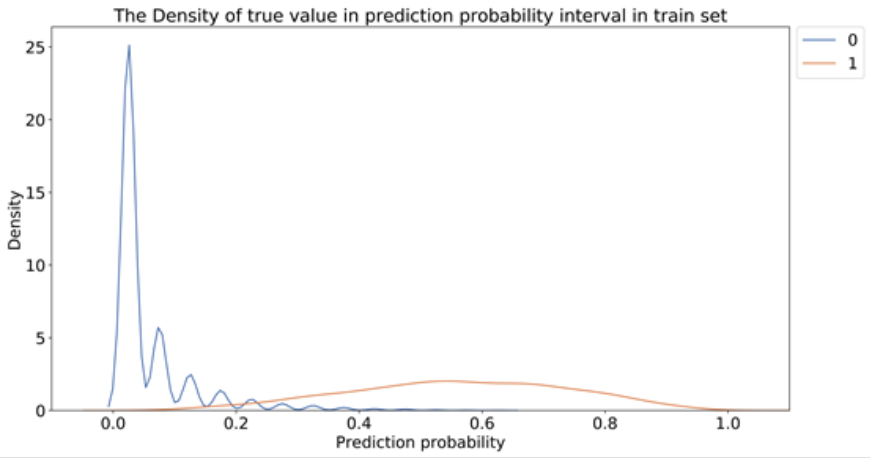

图 4-2 对训练集进行预测结果

我们发现在预测概率大于 0.25 后，真实值为 0 的数量很少，而真实值为 1 的数量开始显著增加，于是，将阈值设定为 0.25，之后开始进行更多的特征工程。

在最后阶段，我们尝试对阈值进行微调，发现在第三位小数上的轻微调整也会造成更大的影响，这是由于在 0.25 附近有较多的概率值。所以我们最后决定不做调整，使用 0.25 来作为最终的阈值。

4.3 线上结果加权集成

当我们通过调参等办法，进一步提升模型，达到 0.406 时，已无法再通过调参来使得模型效果进一步增加。我们试图使用 Stacking/Blending 的常用后处理手段，来提升模型效果，但结果不尽如人意。最后，在发现通过调节训练次数，可以提升结果后，我们考虑使用线上结果加权集成的方法。

该方法可以理解为一种拟合线上分布的方法。通过每个结果的线上成绩进行加权，线上分数高的权重就高，线上成绩相对低的权重就低，最终将加权的结果作为最终的结果(需保证所有的结果的权重之和为 1)。

表 4-1 加权模型

模型	基础参数	num_boost_round	单模型得分
model1	'max_depth':7	500	0.397
model2	'learning_rate':0.01	700	0.401
model3	'num_leaves':50	800	0.402

最终我们选用 0.09 * model1 + 0.15 * model2 + 0.76 * model3，进行融合。

最终得分: 0.4138。

5 创新点与总结

我们将此项目采用的较为新颖的方法和思路总结如下：

1.对缺失值的处理，在比较均值、中位数、众数、0 填充后，我们选择采用训练集保留-99，而测试集将-99 替换成 nan 值，经过验证，无论是在本次比赛，还是在之前四川省金融建模比赛，通过此法进行处理，线上成绩均要高于其他方法处理。

2.特征构造，通过对数据集的探索，寻求到一些关键特征，并正确采用了恰当的处理方法，使得模型性能提升。但可以关注到的是，图 3-6、图 3-7 上所显示 Gain/Split 较为大的特征，在实际线上提升效果并不显著，可能原因是我们设置阈值问题，在加入该强力特征后，我们模型实际预测的概率值上升，我们通过后续新加特征，使得概率值降回 0.25 附近，因此成绩才得以提高。因此，在处理类似问题时，可以积累的经验是，在添加特征后，应观察特征的重要程度，尝试调整阈值（对于此类，手动划分 0-1 概率的数据集），在进行提交，这样可能得到的最终模型，会有更好的提升。

3.在第 4 章中，我们已经讨论到，对于训练集和测试集分布不一致的数据集来说，交叉验证是无法产生较为有用的效果。我们结合得分公式，数据集特点（负样本少，而负样本预测正确与否却极其重要），探索性的使用测试集 1 预测出的数量，与之前最优模型进行对比，来使得模型特征构建逐步推进。

参考文献

[1] Guolin K, Qi M, Thomas F, et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 3149–3157.

[2] Chen T Q, Carlos G. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2016: 785–794.