“魔镜杯”风控算法大赛模型说明文档

全民狙击团队　路丹晖

2016 年 4 月 18 日

目录

1. 赛题分析

拍拍贷“魔镜杯”风控算法大赛开放真实的借贷历史数据,选手需要通过对标的中借款申请人的申请信息、行为记录、第三方征信等数据进行分析，预测标的 6 个月内的逾期率，从而有效的评估借款人借贷违约风险，为授信、风险定价等决策提供依据。

1.1. 比赛数据

比赛提供的数据，其维度主要包括三个部分：

1) 标的借款人的各项数据：用户基本信息、网络行为、学历信息、第三方数据、社交网络数据（总共 226 个字段）;
2) 借款人登录及操作记录（包括操作时间、操作类别和操作代码）;
3) 借款人修改信息的记录（包括修改时间和修改内容）。

复赛的训练数据 79,999 条，测试数据 10,000 条。其中，训练数据中包括了标的的违约标签（1=贷款违约，0=正常还款）。

1.2 数据注意事项及挑战

赛题及数据的基本分析及注意事项包括：

1) 开放的数据字段经过**脱敏处理**（除借款人修改信息外），对于数据的理解及特征构造存在一定的难度，但**依然有一定的空间**；
2) 由于题目数据是拍拍贷真实的标的数据，因此推测这些标的本身是通过了当时的拍拍贷风控审核的，所以在构造特征及筛选特征时，需要充分考虑到标

的的**借款人实际已经通过风控审批，并不是真实的借款申请样本（灰度样本）**，对于变量及其解释性的评估造成一定挑战；

3) 比赛数据中并**无明显重复借贷数据线索**。考虑到网贷客户重复借贷率高的特点，通过比对借款人登录操作及修改信息记录以及包括地理位置等一些用户基本信息的分析，并未发现同一用户存在明显的不同标的，因此考虑比赛数据中已先期去除重复借贷的标的。

2. 模型设计

比赛要求通过历史数据预测标的 6 个月内是否逾期，因此这是一个非常典型的二分类的预测问题。在初赛中，在对给定 29,999 条训练数据进行初步的清洗后（分类变量 encoding/one-hot encoding，缺失值填充），我们利用主流的分类模型尝试 quick model 以确定选型，模型效果如表 1：

表 1 Quick model 试验确定模型选择

Model	**2-fold CV AUC**
logistic regression	0.6832
random forest	0.7011
gbdt	0.7354
svm w/ pca	0.6951
xgboost	0.7467

通过试验发现 Xgboost 和 GBDT 的 quick model 效果相对其他方法更优，且根据以往经验，xgboost 在剪枝、正则化、generalization 方面较传统 gbdt 有非常明显的提升，尤其是算法实现优化后的运行速度提升极为明显，对于模型

开发，算法迭代有非常显著的帮助，所以在我们在比赛初期便选择 Xgboost 为主要模型训练方法。

因为接下来的数据清洗和特征工程，很大程度上依赖于模型自身的特点，由于 Xgboost 基于 Boosted Tree 实现，因此在构造特征时也充分考虑构建适合树形模型的特征变量，例如 dummy，binning 等特征。

3. 特征工程

特征工程是整个模型及算法中最为重要的一部分内容 ,直接决定模型预测能力。这个项目中，整个特征工程分为三个阶段：数据清洗、特征提取以及特征选择。

1) 数据清洗

原始数据中，除了借款人修改信息及登录操作信息外，存在大量脱敏的数据字段，其中包括 Categorical 和 Numeric 的变量类型,并且存在缺失值。绝大多数模型中不能存在字符型变量以及缺失值，因此，在数据清洗阶段，需要完成变量的数值编码、缺失值填充等，具体包括以下清洗环节：

1. 消除包括英文大小写产生的重复值(UserupdateInfo)，中文城市中类似“深圳”和“深圳市”产生的重复值等；
2. 根据变量取值重新定义 Categorical 变量和 Numeric 变量

   除了字符型变量被认为是分类变量外，原始数据中给定的数值型变量的取值小于 20 种时，将被归纳为分类变量，这是基于 Tree model 生成 Tree 的灵活性考虑；

3. Numeric 变量的缺失值被单独赋值填充，通过检验变量与 target 的相关性，选择具体填充的数值，尽量保证缺失值填充后目标变量在 Numeric 变量上的单调性；
4. Categorical 变量中取值超过 20 个的，则直接使用数字编码。唯一的例外包括 UserInfo_2、UserInfo_4、UserInfo_8、UserInfo_20 等城市级别变量以及 UserInfo_24 地址级别变量，这些变量由于取值过多，且每个分类取值中记录条数较少，所以不直接进行编码，将会在下节详细介绍其特征提取方法。而对于取值小于 30 个的，则进行 one-hot encoding 将变量转换为 dummy 变量，将有利于包括回归及决策树相关模型的训练。

2) 特征提取

特征提取是整个特征工程中最重要的部分，决定模型效果的关键步骤。虽然绝大多数用户数据已经进行脱敏处理，对特征提取造成比较大的挑战，但仍然是模型能否取得突破的关键。整个项目开发中特征提取及构建的工作比重最大，也总结出了一系列特征提取的方案，下面将依次介绍通过对数据分析得到的一些发现，并由此构造的一系列独特的特征。

首先，对于整个比赛提供的数据，我们有一个基本的特征提取维度的方案，如图 1 所示，整个特征工程针对 4 类数据，分别对应：

1) 用户修改内容历史
2) 用户登录操作历史
3) 用户基本信息、借款标的信息、社交网络数据及第三方数据
4) 外部数据补充

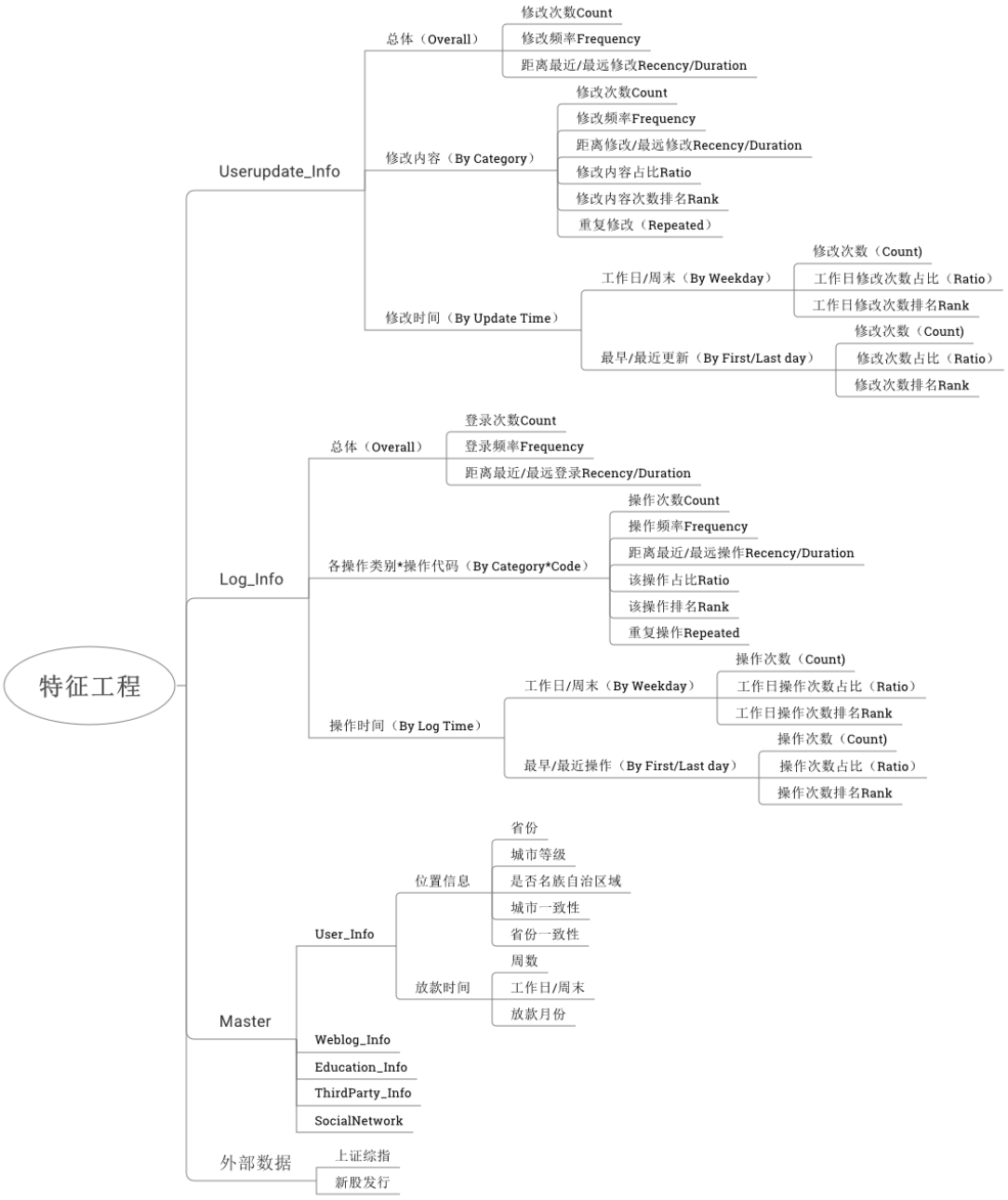

图 1　特征工程概述

下面依次介绍四个部分的特征提取：

1) 用户修改内容

用户修改内容历史记录了用户的修改时间和内容，内容主要包括用户的基本信息（手机、地址等等）以及和用户还款能力相关的数据（教育、是否有车等）。通过分析用户修改内容的行为，我们发现很多与逾期相关的 indicator，这里我们从中列举一些 finding，从而引出针对这部分数据我们是如何构建相关特征的。

a) 用户修改内容的数目，通过图 2 可以发现，超过 38%的用户只修改 12 项内容，且这些用户的逾期率较低，高于或低于 12 项修改内容的用户的逾期率均有上升；

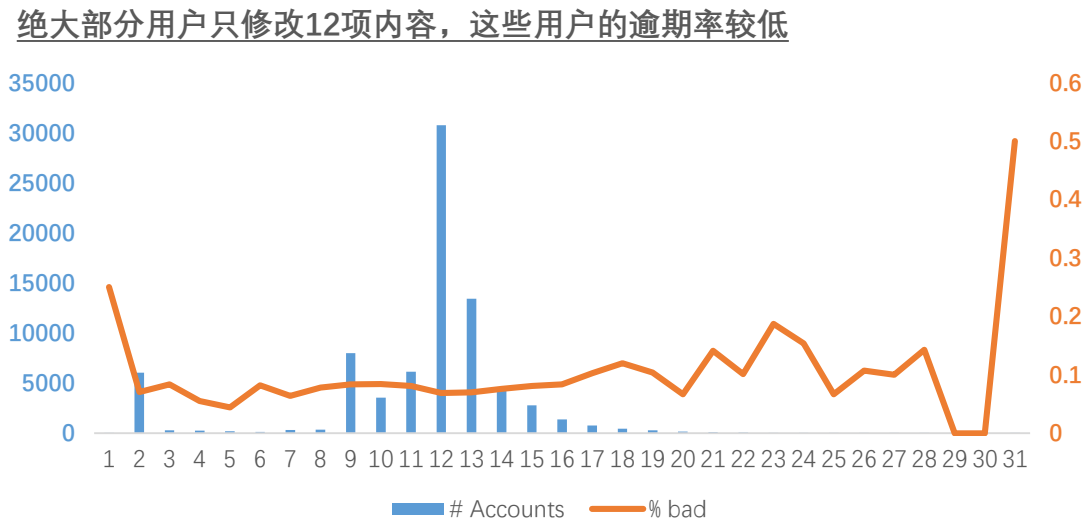

图 2 用户修改内容项数目与逾期率的相关性分析

# update category	# bad	# Accounts	% bad
<12	2,008	25,355	7.92%
=12	2,113	30,758	6.87%
>12	1,829	23,878	7.66%
Grand Total	5,950	79,991	7.44%

Note: 总共只有 79,991 个用户有 User update 数据

b) 同时我们发现用户修改内容的频率越高，逾期率也越高，如图 3 所示。因此，修改频率是一个反映逾期的重要 indicator。

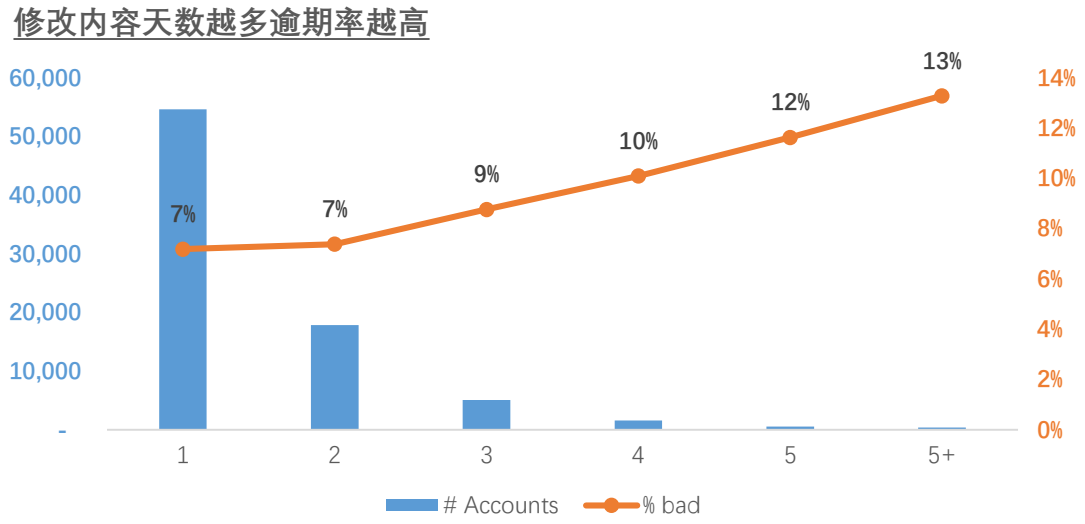

图 3 修改内容频率与逾期率的相关性分析

c) 对于某些特定的修改内容，如电话、QQ、婚姻状态等信息**多次更新**的用户，其逾期的概率也更高，如图 4 所示：

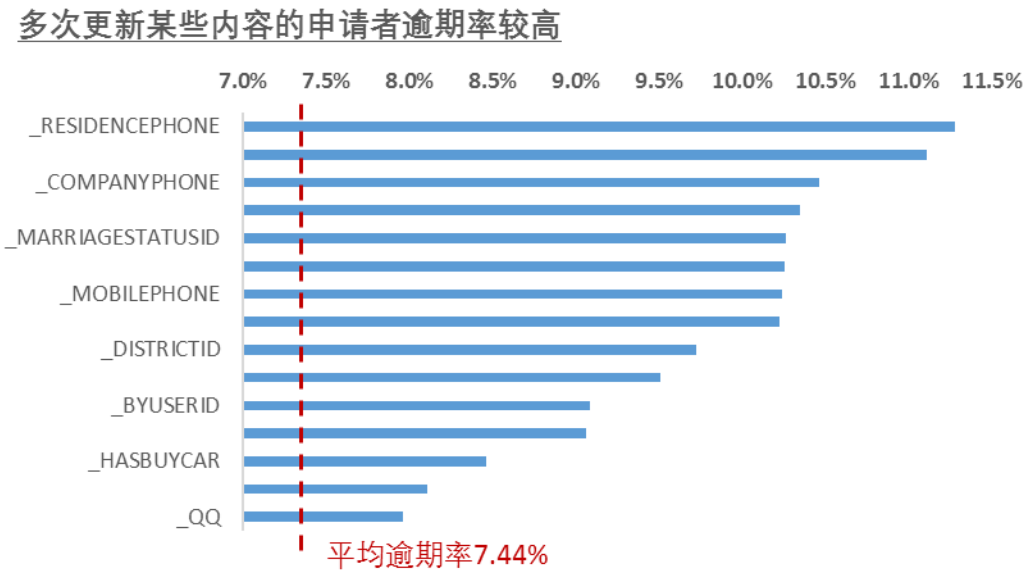

图 4 多次更新内容与逾期率的相关性分析

综上所述，通过对用户更新内容的 profile，我们发现更新的 Count/Frequency/Recency 等均具有一定的 predictive power，因此对用户修改内容的数据，我们提出以下构建特征的方案和角度，如图 5 所示：

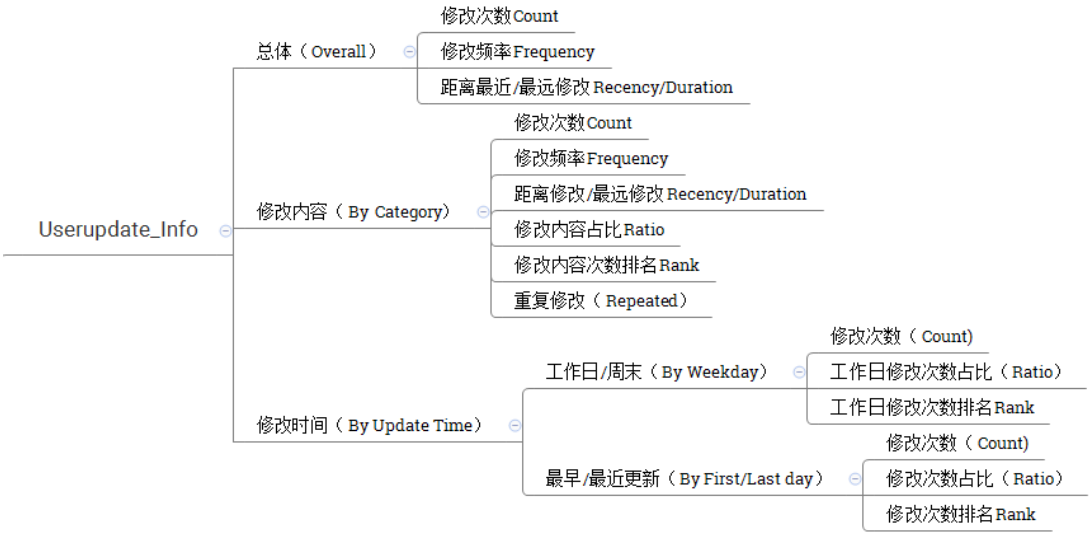

图 5 构造用户修改内容的特征方案

例如，对于一个用户，我们既考虑抽取其总体修改信息的频率、次数等特征，也针对特定修改内容进行特征提取。此外，我们也针对修改的时间进行特征提取。图 5 所示的是关于用户修改内容的特征的主体以及基础维度，实际构造特征时还需要引入时间窗口以及聚合函数，构建丰富的特征，如下例所示：

由于数据量的缘故，在这个比赛中，时间窗口的选择仅为 1/3/7/15/30/ALL 天等，聚合函数包括以下几种：

聚合函数	备注
TOT	Sum
AVG	Average
STD	Standard deviation
Unique	Count Distinct

MAX	Maximun
MIN	Minimun
BUP	Building up
MSN	# of Period Since last not null value
MSZ	# of Period Since last value greater than 0
MSX	# of Period Since largest value
NUM	# of Period with not null value
NUZ	# of Period with value greater than 0

2) 用户登录操作

用户登录操作的特征提取方法基本与用户修改内容一致，如图 6 所示：

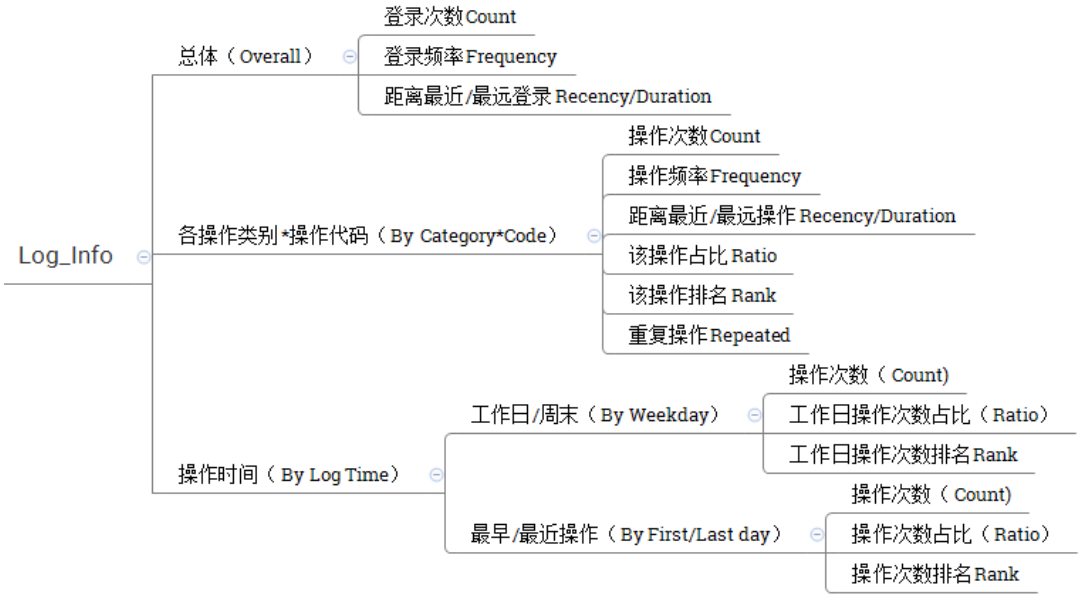

图 6 用户登录操作数据特征提取方案

3) 用户、标的申请数据及第三方数据

这个部分的数据全部进行了脱敏处理，因此除了基本的数据清洗以外，对于

额外提取特征的难度极大。通过观测数据，具有特征提取价值的字段主要有地理信息、第三方信息、时间信息等字段，特征提取方案分别如图 7 所示：

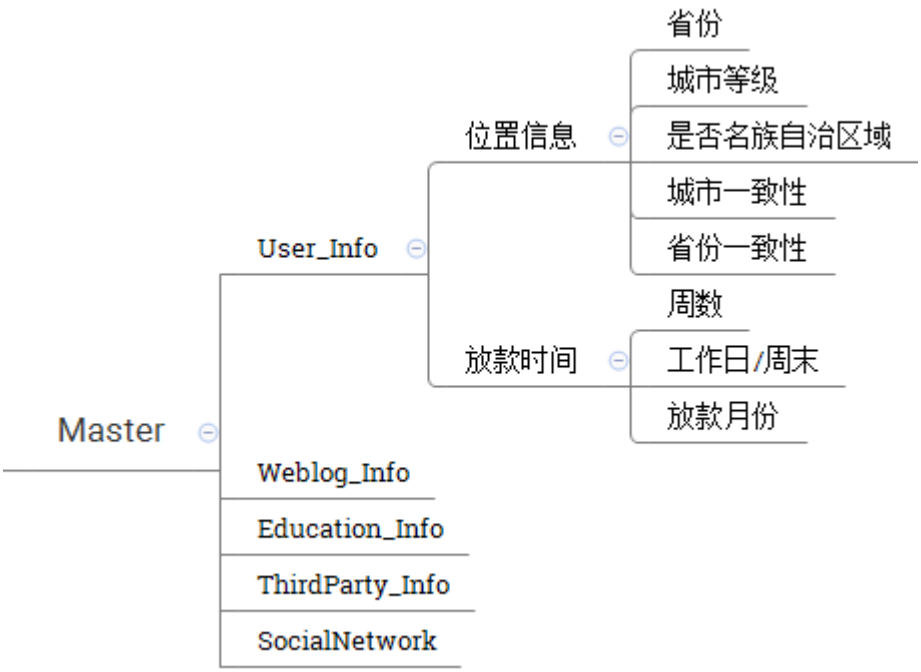

图 7 用户基本信息、社交网络、第三方数据等特征提取

a) 地理信息

地理信息总共涉及 5 个字段，分别为 UserInfo_2、UserInfo_4、UserInfo_8、UserInfo_20 以及 UserInfo_24，其中前 4 者为城市信息，UserInfo_24 中包含少量地址信息。由于 UserInfo_24 中具有地址数据的值仅占 6.9%，所以考虑合并处理。

对于 UserInfo_2、UserInfo_4、UserInfo_8、UserInfo_20，由于城市取值范围非常广（超过 670 个），因此不能直接进行 one-hot encoding，所以先通过外部城市与省份对应数据，将城市 mapping 到对应的省份中，对省份做 one-hot encoding 就较为可行。如图 8 所示，不同省份的借款人逾期率有一定的区分。同时，我们还尝试将相近逾期率的省份进行合并，从而提高模型的泛化性能。

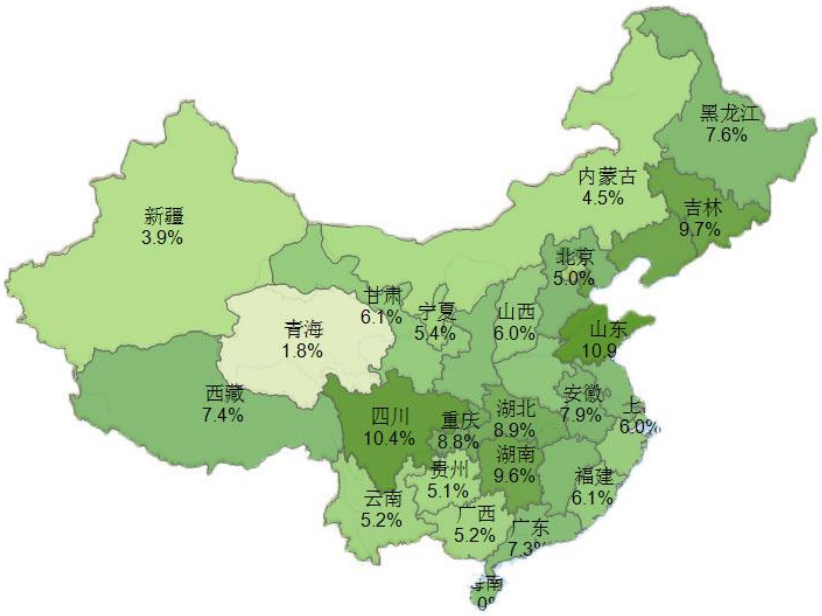

图 8 用户违约率的省份分布

此外，我们发现当借款人的地址信息有重合时，逾期率反而高，如表 2 所示。这里考虑到样本是经过拍拍贷原有风控审核的，因此不能简单的认为这个特征与常理相悖，由于比赛数据信息的缺失，这里只能依靠现有数据的表现构建相对合理的特征。

表 2 位置一致性特征提取（User_Info4/User_Info20）

4_20 same	# non_bad	# bad	总计	% bad
FALSE	41,150	3,030	44,180	6.86%
TRUE	32,898	2,921	35,819	8.15%
总计	74,048	5,951	79,999	7.44%

表 3 位置一致性特征提取（User_Info8/User_Info20）

8_20 same	# non_bad	# bad	总计	% bad
FALSE	43,106	3,193	46,299	6.90%
TRUE	30,942	2,758	33,700	8.18%
总计	74,048	5,951	79,999	7.44%

b) 第三方信息

ThirdParty_Info 字段总共有 17 组，分别对应 7 个 Period，据此认为 17 组 ThirdParty_Info 数据，每组有 7 个时间窗口（并且在时序上存在先后顺序），所以根据上文所述的特征构造方法构造基于时间窗口聚合的第三方数据特征。

c) 时间信息

数据中有放款时间字段，由于数据样本取自 201311 至 201411 超过 12 个月，如图 9 所示，标的的数量逐步上升，而逾期率却保持稳定走低，考虑两个原因：第一，外部经济环境影响，这部分我们会在下节讨论；第二，拍拍贷本身风控模型、策略以及业务的扩张影响。而无论哪种原因作用都会表现在不同时间借贷的人群的表现中，所以这里我们以 Vintage 作为划分，期望能够将风控及业务的影响通过不同 Vintage 的申请人的不同表现所表征。因此，这里我们提取了放款时间的月份、周数等特征。

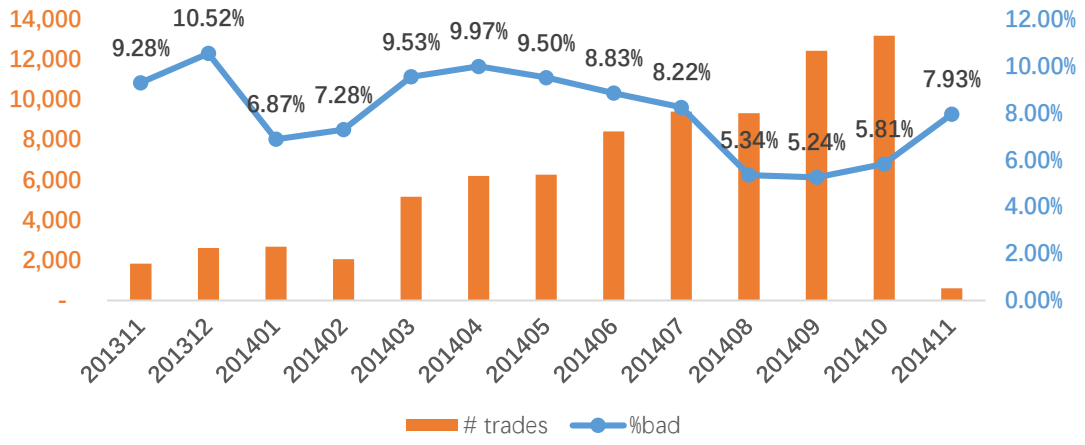

图 9 By Vintage 的逾期率表现

4) 外部数据补充

个人用户的借贷势必受到外部经济环境及政策的影响，包括房地产政策、金融环境变化、银行等机构融资政策的变化等等。除此之外，对于个人而言，最直接的影响因素即为股市。通过调研发现 2013 年 11 月至 2014 年 11 月间，A 股

市场具有两个很明显的变化。

第一，从 2014 年下半年起，A 股进入一个比较稳定的上升走势；

第二，2014 年 1 月起，IPO 暂停后首次重启。

通过数据也发现，这两点股市的变化均与用户逾期存在一定的相关性，如图 10 所示，如果不考虑 2014 年 11 月的逾期数据（样本较少），在 201408~201410 股指上行的 3 个月中，逾期率下降明显。因此我们依据放款时间，构造该时间点及其历史的股指特征。

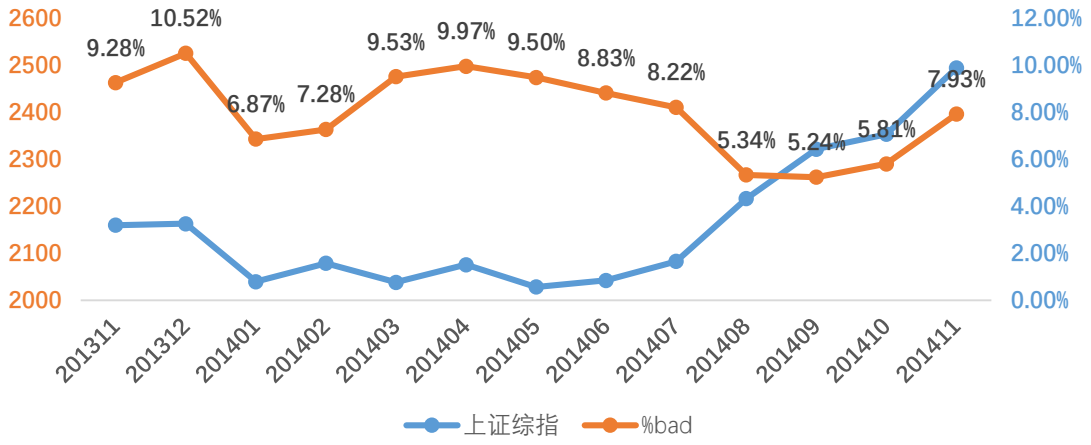

图 10 上证综证与逾期率的相关性分析

继续观察 IPO 解禁后对于逾期率的影响，2014 年 1 月 IPO 重启，大量新股申购，众所周知中国股市新股破发概率极低且平均收益接近 30%，所以如果拍拍贷客户此时的资金流向是申购新股，则资金风险本身就非常低，进而逾期概率也会低，可以参考图 11 至图 13。新股申购时段的逾期率均低于其他月份，所以我们认为新股的申购对与逾期有一定相关性。

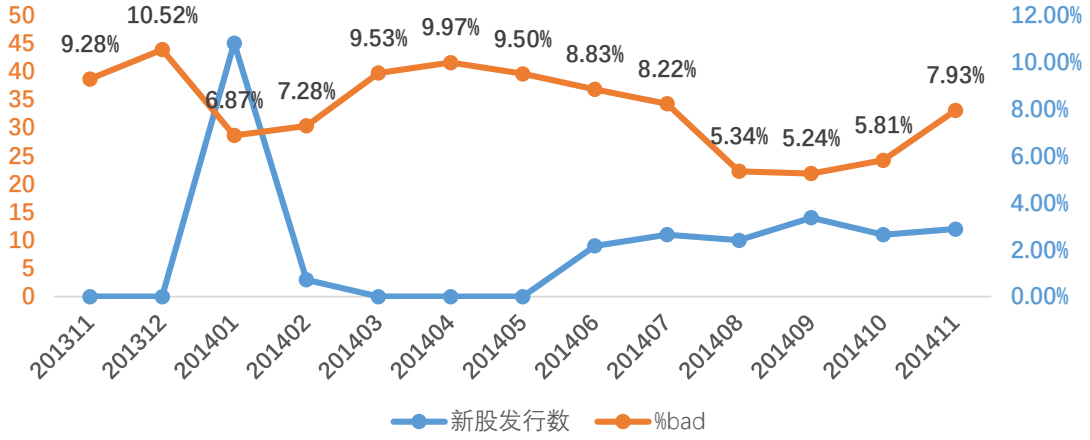

图 11 新股发行数与逾期率的相关性分析

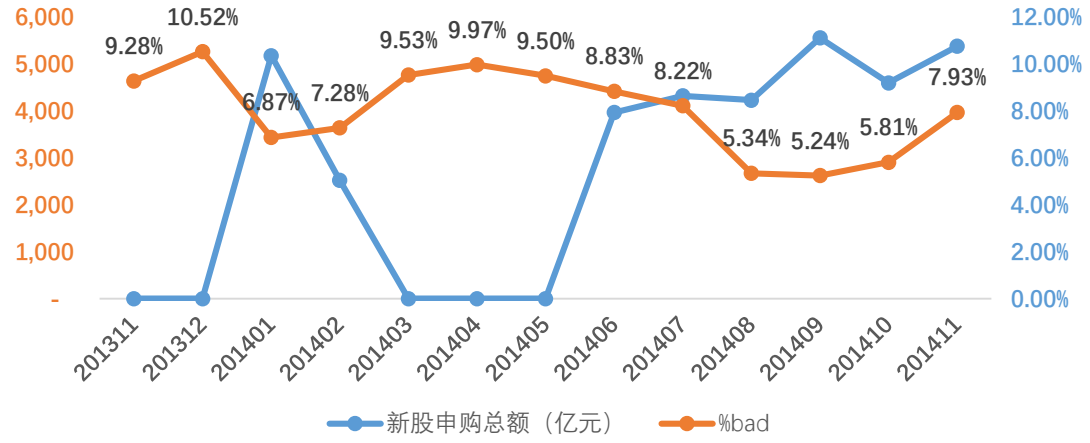

图 12 新股申购总额与逾期率的相关性分析

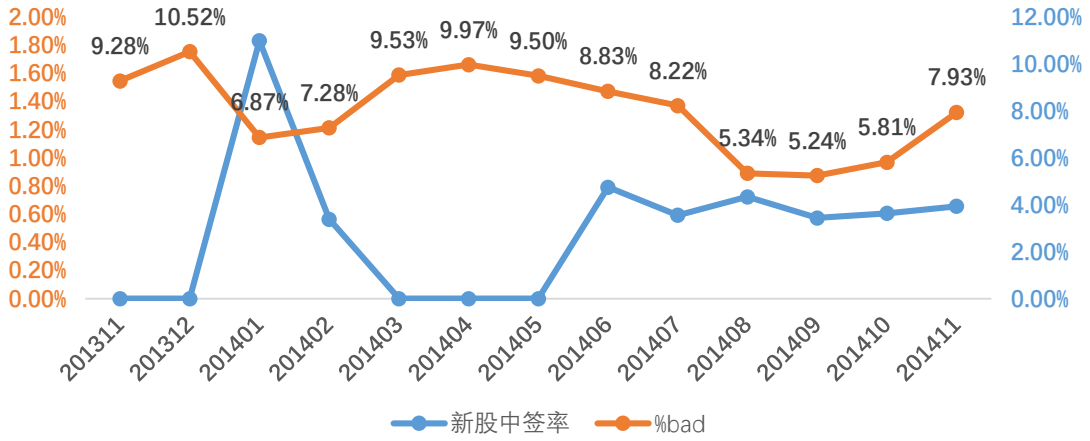

图 11 新股中签率与逾期率的相关性分析

综上，我们引入放款当日的股指，以及当天/未来 3/7/15/30 天内新股申购的相关特征（新股发行数、申购额、中签率等），由于一个月内新股申购的信息属于公开信息，所以该数据可用于实际工程中。

3) 特征选择

通过上述的数据清洗及特征提取方法，我们一共生成了 2865 个特征。接下来需要对特征进行筛选，从而提升模型训练的效率和精度。特征选择的流程如下：

a) 将训练数据随机等比例划分为训练集和验证集，删除 **0.5%至 99.5%峰位内 variance 等于 0** 的变量；

b) 使用 Xgboost 做一次 10Fold 交叉验证，将十次训练的模型的 feature importance 取出 取 10 次中超过 5 次出现在模型并且出现 10 次 boosting 以上的特征组合，删除剩余特征，保证模型在不同训练样本的稳定性；

c) 对数值型变量以及 dummy 变量计算 information value，将排名前 100 的变量加回到特征库中；

d) 在特征库中计算 Pearson correlation，删除绝对值超过 0.95 的特征组合中的一个；

e) b~d 步根据模型交叉验证效果进行迭代调整；

f) 最终得到 510 个特征进行模型训练.

4.模型训练与评估

4.1 样本设计

复赛中提供 79,999 条训练数据，预测 10,000 条测试数据，模型训练及调参时，为兼顾效率和评估准确性，主要使用 2-fold 交叉验证。参数基本确定后使用 10-fold 交叉验证进行参数调整和进一步评估以及确定最终的迭代次数。所有的 k-fold 样本均使用 **target 进行分层抽样**，保证不同 fold 的 auc 评估具

有可比性。

4.2 模型训练

使用 Xgboost 训练模型时，最重要的参数的调整，在实际操作中，需要理解模型 Bias-Variance Trade off 进行参数的调整，本次项目的调参主要借助 sklearn 中的 grid search 进行参数的逼近，主要步骤如下：

a) Learning Rate/number of boosting round。首先为了加快模型训练速度，调整较大的 learning rate，使得每次 boost 的权重增加，因此需要 boost 的次数就越小，所以调参初期选择 learning rate 在 0.05~0.1 之间；
b) 在一定的 learning rate 和 boosting round 下调整模型的 max_depth 以及 min_child_weight，前者是每颗数的最大深度，深度越大 bias 减小，同时 variance 增加，min_child_weight 是每个叶子节点的最小观测数目，该数目越小，则 bias 越低。所以这两个参数要一起调整并确定。
c) 调整 subsample 以及 colsample_bytree，前者是每次构造 tree 时随机取样的比例，后者则是随机选择一定比例的特征，减小这两个比例能够增加随机性，从而降低 variance。
d) 最后调整正则项，并且返回步骤 a，逐步减小 learning rate 并增加 num_round。

4.3 模型融合

模型融合（Model ensembling）在诸多项目及比赛中，被认为是提升预测能力的有效方法，原因是通过不同模型的组合，降低了 generalization error。

所以在本项目中，也尝试了多种模型融合方案。

4.3.1 Averaging ensemble

Model averaging 在绝大多数模型融合问题中能够取得效果的提升，因为 averaging 主要降低模型的过拟合问题。我们采用将不同参数的 xgboost model 的打分进行平均，得到最终评分，50-50 的训练集及验证集测试结果如图 14 所示，通过简单的 averaging 后，模型提升至少 0.001 点 AUC。

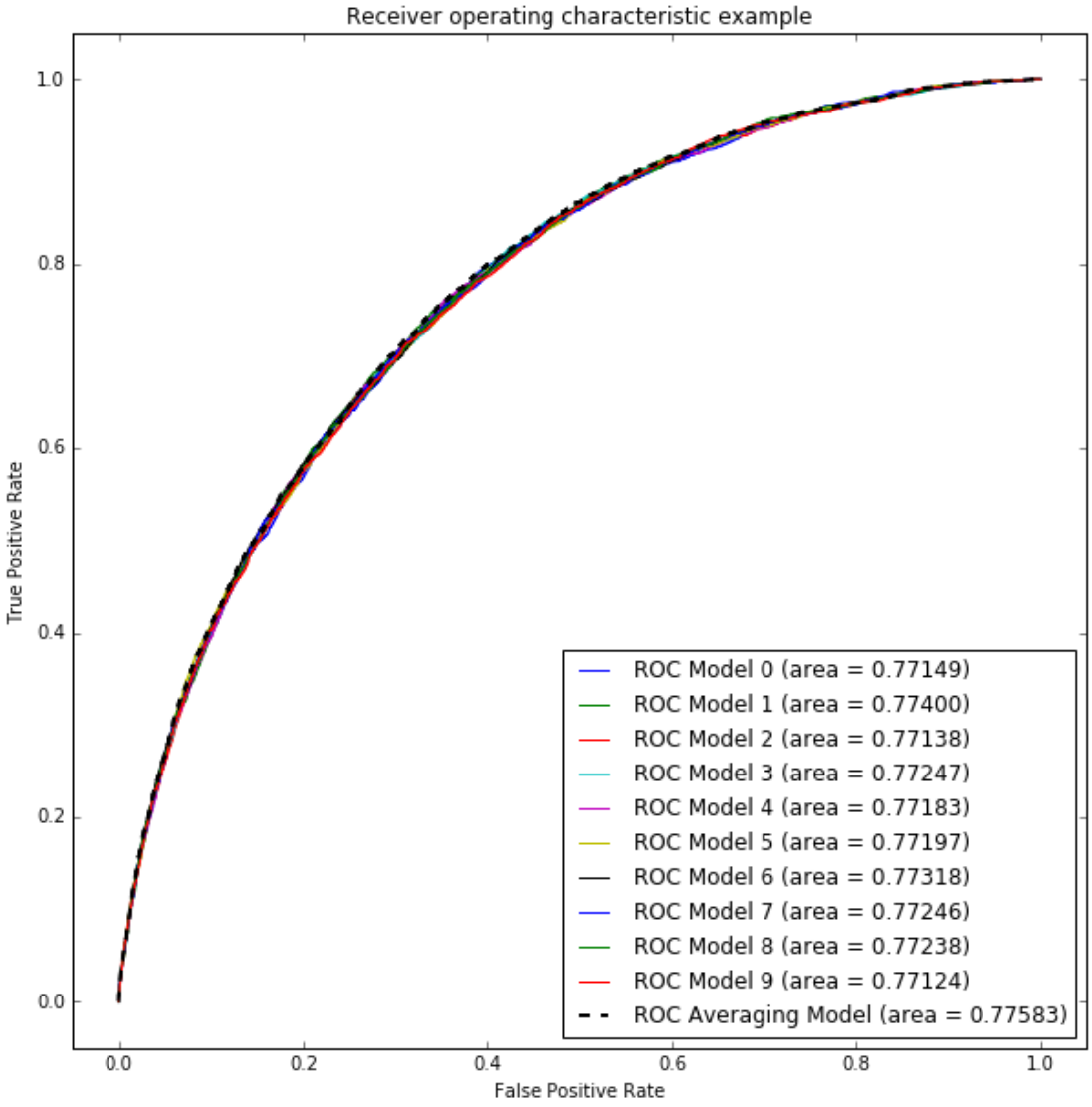

图 14 Averaging Model 的效果提升分析

由于 Averaging ensemble 思路相对简单，且模型不存在级联关系，兼备准确性和稳定性的特点，模型的仅由常规的 boosting tree 组成，复杂度较低，

算法实现简单且单次训练时间短，因此最终的模型融合方案选择此方法。我们选择了 4 个候选的 xgboost 模型，对结果进行 Averaging，得到最终的预测结果。

4.3.2 Stack Learning

复赛中的训练数据较为充足，因此尝试 3 种 2-layer stack learning 的方案，如下图所示，第一层训练 XBG/GBDT/RF 等树形结构的弱分类器，第二层尝试 XGB/KNN/LR 等方法进一步提升分类器精度。

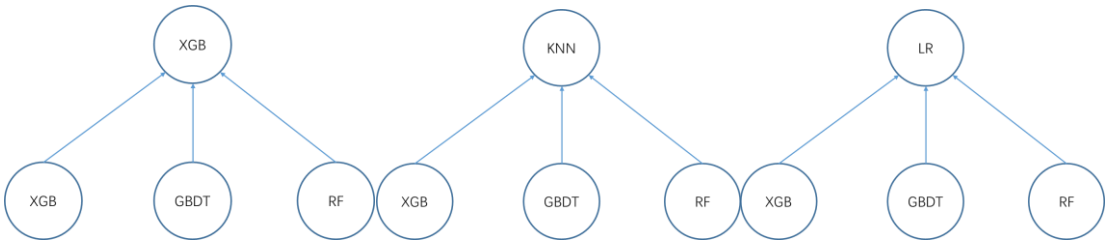

图 15 Stack Learning Schema

10-fold 交叉验证效果如下表：

	Training method	10fold CV AUC	备注
Layer 1	XGB	0.7763	learning rate=0.02 n_estimator=3500
	Random Forest	0.7457	n_trees=2000, max_depth=15
	GBDT	0.7689	learning rate=0.02 n_estimator=2000
Layer 2	XGB	0.7809	learning rate=0.02 n_estimator=2000
	KNN	0.7655	K=100
	Logistic Regression	0.7734	

4.3.3 Stack+Averaging ensemble

进一步增加模型复杂度，融合 stack learnning 和 model averaging，可以进一步提升模型的效果（2-fold cv auc 0.7783），如下图所示结构。但模型的训练成本随之上升，在 i7+16G PC 环境下，510 个特征，2-fold 交叉验证运行时间长达 6 小时 48 分钟。由于模型复杂度考虑，并没有继续尝试此方案。

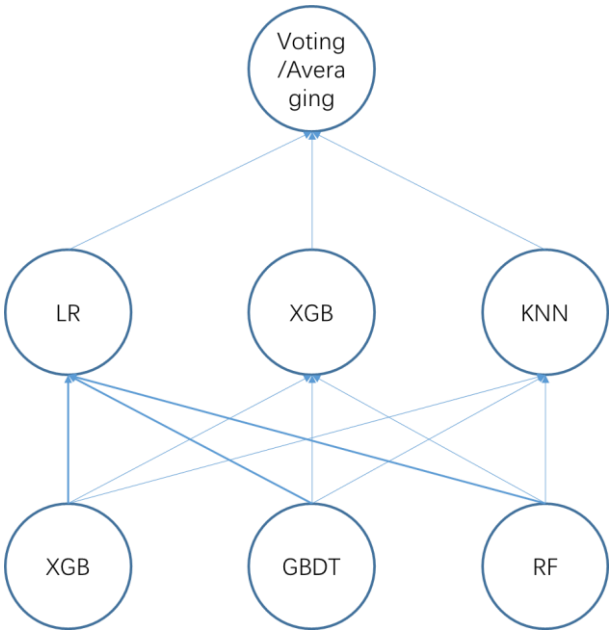

图 16 Stack Learning 与 averaging 融合

4.4 模型验证

比赛使用 AUC 作为模型评判标准，且线上的评测机会非常有限，因此需要构建高效且准确的本地模型验证流程。我们主要采用对目标变量分层抽样的交叉验证方法对模型进行评估，评估的标准与比赛一致（AUC）。由于计算资源限制，且特征维度较多，因此我们使用 2-fold 以及 10-fold 两个级别的交叉验证，前者是对模型效果进行初步的评估比较，优点包括：

a. 模型验证速度快

b. 可以比较不同模型的相对性能，从而判断模型好坏

同时存在两点问题：

a. 验证时训练样本的比例降低，对估计线上的 AUC 的数值参考存在偏差

b. 对整个训练集进行训练时的参数指导

所以在 2-fold 交叉验证选取模型后，进一步进行 10-fold 交叉验证，此时 auc 结果与线上测评时非常接近，基本在交叉验证的 auc 标准差范围内，同时在 10-fold 的结果上，对全部训练集重新训练时，只需要增加 10%的 boosting 的次数即可。下图是一组 10-fold 交叉验证的图表，从该图中可以了解模型的 auc，auc 标准差以及合理的 boost 次数。

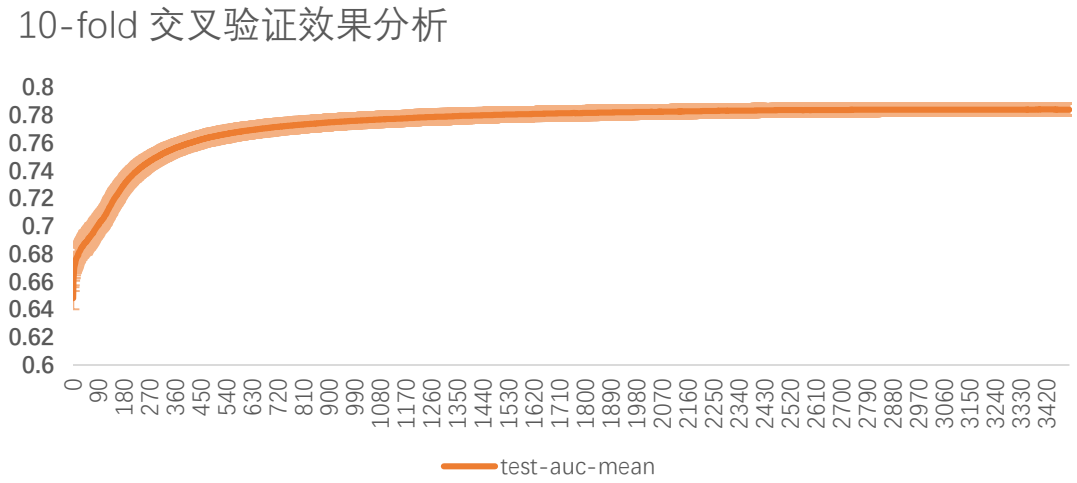

图 17 10-fold 交叉验证分析

5.心得与体会

非常感谢拍拍贷平台以及 Kesci 平台能够提供这样一次非凡的比赛机会。尤其是在外部舆论对互联网金融发展存在质疑的敏感时期，拍拍贷无疑向公众展示了其丰富的数据维度以及强大的风控技术，对于整个行业的发展而言无疑起到的推进作用。作为选手，也在这次比赛中学到很多新的技术、知识，结识了更多的

朋友，并且发现自己的不足。

也希望 Kesci 能够继续不断发展壮大，为国内数据爱好者提供更多的机会和锻炼的平台，衷心感谢所有工作人员的努力，谢谢。