融 360“天机”金融风控大数据竞赛

北京赛区

临兵斗列

夏丹青 缪世磊 刘羽中

2016/10/08

内容目录

一、队伍介绍

队伍“临兵斗列”由三位小伙伴组成，我们是在数据城堡 DataCastle 平台上的《猜你喜欢》比赛上认识，并在大数据挖掘的爱好趋势下一起走来。队伍成员信息如下：

姓名	学校/公司、学历	比赛经历
夏丹青	北京航空航天大学 --- 研二	DataCastle- 猜你喜欢第四名，数据挖掘爱好者
缪世磊	万达信息--- 研究生	DataCastle- 猜你喜欢第四名，从事相关工作
刘羽中	北京航空航天大学--- 研二	DataCastle- 猜你喜欢第四名，机器学习爱好者

二、比赛解决方法概述

2.1 比赛问题分析

融 360 平台的一端是亿级别有借款需求的小微企业和个人消费者，另一端是有贷款资金的万级别的金融机构，平台通过搜索和推荐服务来撮合借款用户和贷款。在此过程中，用户通过平台的搜索和推荐服务查找适合自己的贷款产品，并填写资料，提交贷款申请，金融机构通过我们的平台收到订单后，对用户资质进行风控审核，决定是否通过用户的订单。

比赛提供了去除用户信息后的四类数据：

- 用户基本信息数据：用户平台上提交的基本资料，包含修改记录
- 用户消费信息数据：用户向融 360 平台上提交的消费信息数据
- 用户行为标签数据：用户在融 360 平台的行为标签
- 用户社交关系信息：部分用户社交关系及关系强度。

比赛目的是在给定用户信息的基础上，对用户是否会二次贷款进行建模预测。问题转换成 2 分类问题,评估指标为 AUC,其本质是排序优化问题。

2.2 解决方法思路

对于本赛题，本队从主要一下三个方面进行分析、建模的。

1、探索数据表的分布，主要包括数据维度分析、数据缺失分布情况分析以及其他的一些表整体的描述性统计分析；

2、特征工程，对用户基本信息表、消费表、用户社交关系表和用户标签表构建特征；并结合特征选择（本队采用的 xgboost 算法对生成的特征进行重要性排序，并根据 train/test 部分表数据分布不均匀等情况进行特征剔除），构建最终模型所需要的特征。

3、建模，采用 Xgboost 算法、RF 算法以及配合 C4.5 结合使用；最后进行模型融合。

4、利用不同算法的预测结果，对预测结果进行融合。

三、数据分析

比赛总共提供的数据表有 7 项：user_info 数据表为用户信息表，consumption_recode 数据表为用户消费表，rong_tag 数据表为用户行为数据表，relation1 数据表和 relation2 数据表为用户社交关系信息表。以及带有训练 lable 的 train 训练表，和需要预测的 test 数据表。

由于各数据表之间 user_id 分布不均匀，存在多个数据表训练部分和测试分布不一致的情况，需要提前对数据整体的分布进行探索分析，为特征选取工作和本地评测环境提供指导方向。

3.1 数据整体分布

数据整体分布情况如下：user_info 用户基本信息表，包含训练集和测试集出现的所有用户；consumption_recode 消费信息表，包含训练集和测试集出现的部分用户；rong_tag 标签数据表，包含训练集和测试集出现的部分用户；relation1 和 relation2 用户社交关系相关信息表,包含训练集和测试集出现的部分用户的关系类型及强度。数据表之间用户在训练部分和预测部分分布不均匀，整体用户数量分布图如下图 3-1 所示。

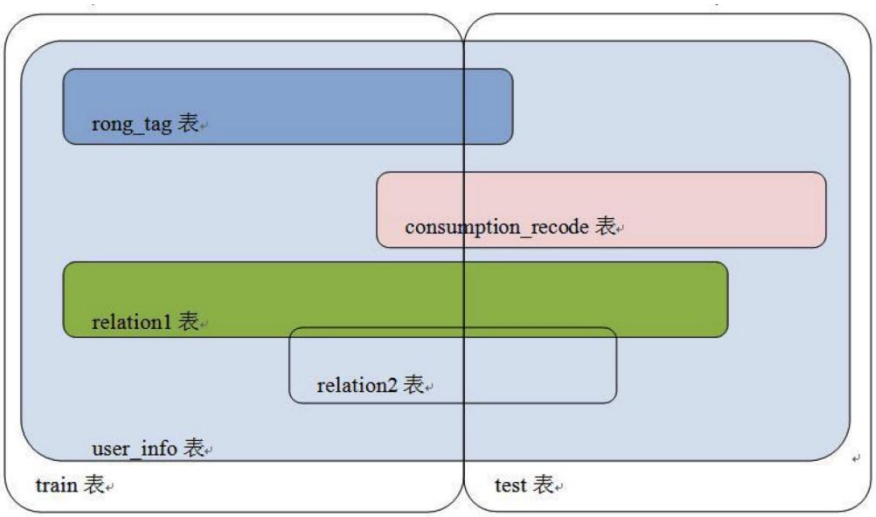


图 3-1 数据整体分布图

3.2 其他表的描述性统计分析

3.2.1 user_info 用户信息表

user_info 用户信息表包含有所有的用户个人基本资料，如年龄、性别、职业、教育信息和个人经济情况等，包含修改记录。user_info 用户信息表有以下关注点：

1. user_info 表包含所有的 user_id 基本信息，是所有信息表中覆盖最大的表。
2. user_info 表包含用户的基本信息以及包含用户的修改记录和最后修改时间；
3. user_info 表包含了训练集跟测试集的所有用户信息，此时不存在构建模型的训练集与测试集分布不均匀问题。

Step1：

首先探索 user_info 表中每条记录数据项缺失的情况，便于我们进一步掌握 user_info 表的数据分布。将 user_info 数据表缺失项统计分别在训练数据集和测试集上面的分布，如下图 3-2 所示。

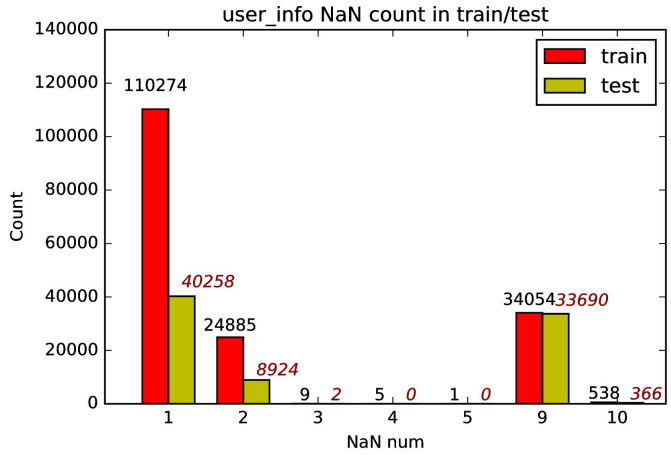


图 3-2 user_info 表缺失项 train/test 分项统计

从 3-2 来看：1）可以剔除缺失值数量为 3/4/5 的记录，减少数据噪音；

2）在缺失个数为 9 的时候，训练集与测试集的样本量比为 1.01：1，很明显这是数据内部结构导致的缺失情况，说明后续需要利用表的数据项缺失这个信息。

Step2：

为了进一步分析表的数据结构，接下来绘制缺失个数与索引之间的分布关系。如图 3-3 所示。

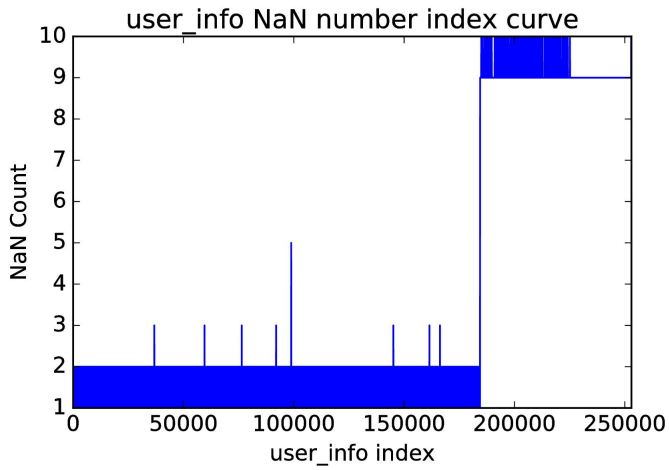


图 3-3 user_info 表记录缺失数索引图

从图 3-3 很明显可以看出：1）user_info 数据表的前半部分主要由缺失值数量 2 的纪录项组成，后部分主要由缺失值数量为 9 的记录项组成；

2）进一步分析发现，缺失个数为 2 的纪录是 product_id=1 的记录组成，缺失个数为 9 的纪录是 product_id=2 的记录组成。

Step3：

user_info 表中数据项缺失的情况，如图 3-4 所示。

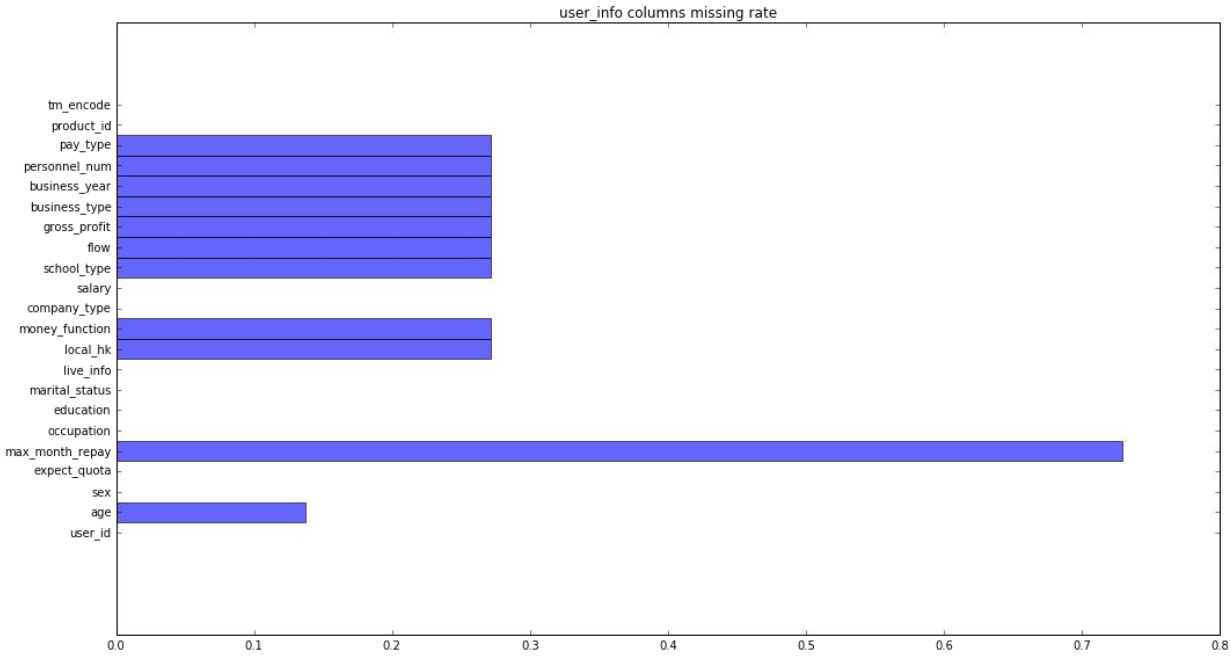


图 3-4 user_info 表数据项比例(横轴比例，最大为 1)

从图可以看出数据项 user_id、sex、occupation、education、marital_status、live_info、company_type、salary、product_id 和 tm_encode 数据项缺失率为 0，local_hk、money_function、school_type、flow、gross_profit、business_type、business_year、

personnel_num 和 pay_type 数据项缺失率相同，max_month_repay 数据项缺失率接近 75%，这是由于不同 product_id 记录缺失项不同导致的。

3.2.2 consumption_recode 消费信息表

consumption_recode 消费信息表包含用户在 rong360 的消费情况，且平均每个用户包含多条消费记录，信息量很大。用户最后的消费状态如 current_bill_bal 本期账单余额和 current_bill_amt 本期账单金额能一定得到用户的消费状态。

3.2.3 rong_tag 用户行为表

rong_tag 数据表是用户行为数据，即用户行为的记录。如下表所示，rong_tag 数据在训练集和测试集中的分布比例不一致。通过分析发现，定义 N_ rong_tag =（rong_tag - 300000)/7 可以将原始 rong_tag 值转换在 1- 37359 范围的数值，并且值越大对应的记录频次越小。N_ rong_tag=300 对应的记录数为 317，因此本队采用 N_ rong_tag 指标的前 300 个进行特征提取。下表 3-1 是 rong_tag 表的基本统计。

表 3-1 rong_tag 信息统计表

		train/26000	test/12261
user_id 数量		12481	4409
tag 数量		30420	16798
user_id 平均 tag 数量		40.7	40.7
train/test 共有 tag	tag 数量	9859	
	tag 比例	94%	96%
热门 50tag 所占比例		35.9%	35.8%
热门 200tag 所占比例		62.0%	61.2%

3.2.4 relation 用户关系表

1.relation1 关系表

relation1 关系表是无方向信息的社交关系表，可以将关系信息转化为无向图图模型，将用户做节点，用户之间的关系看做边，进行数据统计。relation1 表统计信息如表所示。

表 3-2 rong_tag 信息统计表

	train/26000	test/12261
user_id 数量	23041	8906
user_id 一层好友平均数量	240.4	1270.4
user_id 二层好友平均数量	89.8	496.9

user_id 回环数量	5210	1851

2. relation2 关系表

relation2 关系表与 relation1 关系表不同，relation2 是有向图，且关系记录有时间和类型等信息。我们对 relation2 表的处理方法是统计用户各类信息的统计值。

四、特征工程

4.1 特征生成

➢ user_info 个人信息特征

user_info 信息表覆盖了所有的训练集和测试集用户，用户信息特征非常重要。在分析 user_info 信息表过程中，发现用户信息项的缺失情况与用户的 product_id 有关，但为了模型 AUC 排序的一致性，我们还是同一建模和生成特征。我们对 user_info 信息表中用户的信息进行的提取，对多条信息不一致的情况进行了编码，单独作为特征；对类别信息进行 OneHot 编码；对 tm_encode 记录求取均值、方差和差值等特征。我们单独用用户信息特征线上可以达到 0.6 的结果。

➢ consumption_recode 用户消费特征

consumption_recode 消费信息的特征我们暂时只是统计了用户的消费信息，并把用户的最新消费信息单独作为特征。

➢ rong_tag 用户行为特征

rong_tag 数据表中的 tag 标记在训练集和测试集中的分布非常不均匀，通过 3.2.2 节分析后，我们将高频的前 300 个 tag 作为特征按照用户记录进行 One-Hot 编码。此时特征量达到 300，本队采用 XGBoost 算法进行了特征提取，最终进入模型的特征及其重要性如下表 4-1 所示，其中那个-1 代表缺失。

表 4-1 rong_tag 表特征提取结果表

Feature	Importance	n_min	n_max	n_mean
rong_tag_192	1	-1	1	-0.501923077
rong_tag_134	1	-1	1	-0.492192308
rong_tag_155	1	-1	1	-0.497115385
rong_tag_57	1	-1	1	-0.455961538
rong_tag_241	2	-1	1	-0.508

rong_tag_4	2	-1	1	-0.122307692
rong_tag_17	2	-1	1	-0.377307692
rong_tag_13	2	-1	1	-0.369807692
rong_tag_107	2	-1	1	-0.485538462
rong_tag_67	3	-1	1	-0.462807692
rong_tag_31	3	-1	1	-0.423961538
rong_tag_56	3	-1	1	-0.456423077
rong_tag_29	3	-1	1	-0.422153846
rong_tag_79	3	-1	1	-0.469961538
rong_tag_173	8	-1	1	-0.498615385
rong_tag_55	9	-1	1	-0.449807692
rong_tag_53	13	-1	1	-0.450884615
rong_tag_86	15	-1	1	-0.474423077
rong_tag_59	20	-1	1	-0.452
rong_tag_28	22	-1	1	-0.437923077

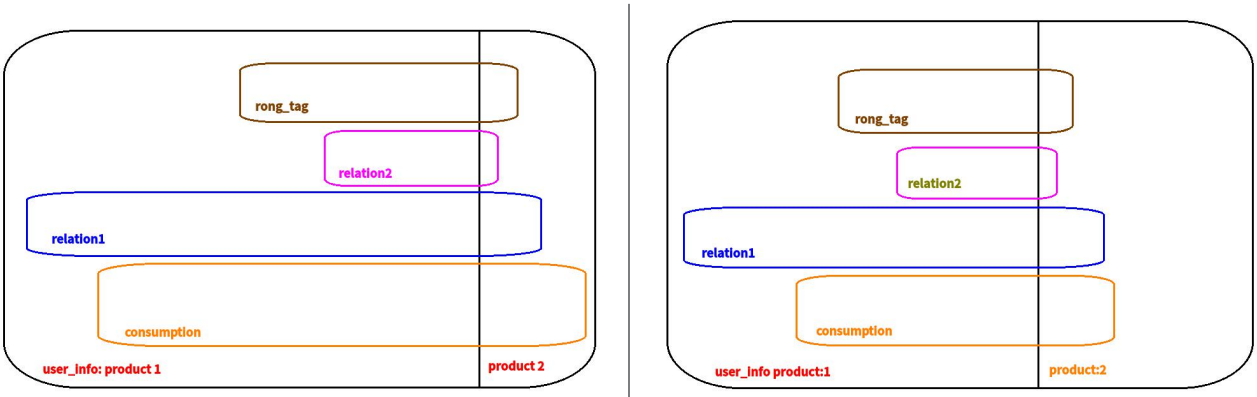
- realtion 社交关系特征

用户社交圈越大，证明用户在 rong360 平台上越活越，表明用户与平台联系紧密程度。我们统计了 relation1 关系表中用户一层好友的数量(直接存在记录的用户个数)和二层好友的数量(好友的好友的数量和)作为用户的社交特征。对于 relation2 关系表，我们统计了一层好友数量和二层好友数量之外，还统计了用户关系的出度和入度，以及关系对应的类别信息。在 relation1 表和 relation2 表中，都存在用户与自身关系的情况，我们把这种用户自身联系提取了特征。

4.2 特征选取

第三步的特征工程，本队初步生成的特征为 310 维，本队主要采用最大信息系数(MIC)、皮尔森相关系数(衡量变量间的线性相关性)、正则化方法(L1,L2)剔除了无关特征，以及结合 XGBoost 算法、RF 算法进行特征提取。

在 3.1 节数据整体分布分析中，我们已经发现各用户表的信息分布不一致，见图 3-1 所示。我们在此基础上分析训练集和测试集各自的分布项，如图 4-1 所示。



(a) train 训练集分布　　(b) test 测试集分布

图 4-1 product_id 类别用户分布图

如图 4-1 所示，可以将数据从训练集/测试和 product_id 两个维度进行划分。可见数据集的分布非常不均匀，从整体上，测试集的用户的数据记录非常稀疏。我们在模型测试阶段也发现了本地交叉验证成绩与线上成绩不符合的情况，也说明了训练集和测试集的数据分布非常不一致。

4.3 异常的剔除

这里的异常主要涉及两个方面：1）异常特征；2）异常训练集样本。

本队最终构建模型进行测试时，总共构建了 307 维特征，但是有些特征在训练集跟测试集的分布情况差异非常大，因此本队主要剔除了 rong-tag 表提取的特征，只保留了最重要的三个特征加入最终模型构建的特征里面。

此时特征个数为 290 个，结合上面的分析，本赛题的样本特征缺失个数跟 lable 有很强的关系，这也从某个层面反映了数据的结构分布，下图 4-2 为统计的结果。

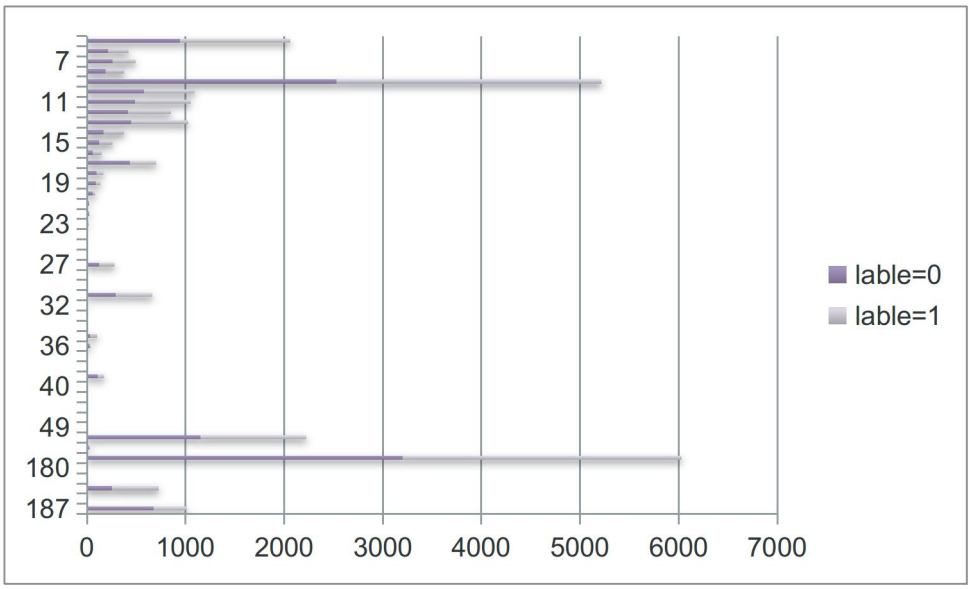


图 4-2 样本特征缺失个数跟标签（lable）的关系图

上图可以看出，在用户特征缺失个数为 187 时，存在的样本个数为 1018，此时的 lable=0：lable=1 与其他的缺失情况出现明显的反差。通过分析发现，此时完全为 product_id=2 的样本。

因此最终构建的模型，本队通过线上跟线下测试，最终选择剔除用户特征缺失个数为 187 的样本，为 1018 个，此举让线上 AUC 提高 0.0018。

五、模型设计与分析

5.1 本地测试环境

为了验证模型的参数的有效性，我们在本地设计了两种验证环境：

- 十折交叉验证：使用 sklearn 的十折，对训练数据集进行拆分
- 时间差验证：将训练集从用户的 tm_encode 进行拆分，并在两部分随机采样

5.2 模型设计

我们主要选取了 RandomForest 和 XGBoost 作为比赛模型，进行分别训练。这两个算法各自有各自的优势，本队最终测试的结果线下跟线上也是 XGB 比 RF 效果好。不过本队最终采用的是融合两个模型的结果。

我们还在训练的模型的过程中，在构造特征的基础上，从生成特征缺失的统计，剔除了训练数据集中与测试集分布不均匀的部分数据以及特征。

5.3 模型融合

由于我们特征选取的还不够完全，模型融合步骤做的非常简单。我们首先找到了一个不错的 XGB 参数，然后让其内部参数随机扰动生成 n 个 XGB 特征，与 RF 融合，得到最后的结果。

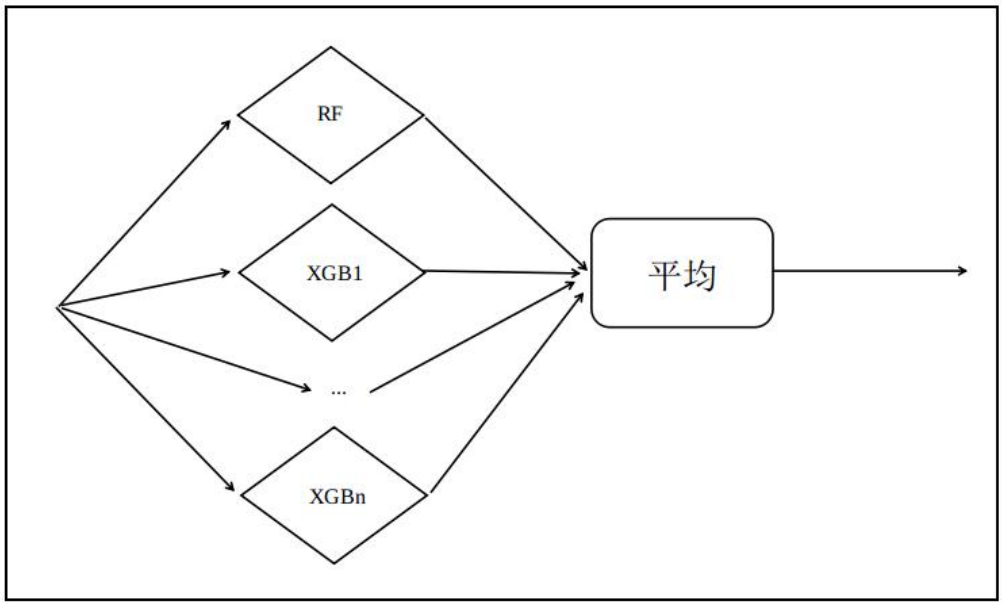


图 5-1 模型融合结构

5.4 AUC 融合

AUC 指标衡量的是预测概率的排序，每次模型预测的结果也对应一个方差。结合指标的构造思想，以及能在一定程度上降低模型的方差（预测值的波动），本队设计了对原始预测概率进行非线性转换的方法，使得本文提交的结果有了一定的提高，线下测试提高点为 0.0006，线上测试提高 0.0002。且保证最终提交的预测值范围分布在[0,1]之间。

```
def rank_transform(data):
    for i in range(len(data)):
        data[i] = math.log(data[i],1.1)
    data = math.log(int(len(data)/2*2.3),1.1) - data
    return DataFrame(data)
```

以上函数是对预测概率的 rank 进行了一个非线性转换，转换后记为 P2，原始概率记为 P1，则最终融合后的预测概率 P3=2*P1/P2，本队的 P3 值预测范围为[0.00433410101103，0.964957273118]。

六、参赛感想与建议

通过本次实际参与风控比赛的经历，让我们感受颇多。与以往的风控比赛不同，我们觉得本次风控比赛从数据上更加贴近真实数据，数据分布不均匀的情况很明显，这种情况直接影响模型的设计以及本地测试环境。精度高的模型并不一定保证好的效果，需要从数据分布上分析特征，进而建立模型，才能保证最终结果。应该从数据分布就直接指明模型的设计，不应该从数据模型特征反过来分析数据分析，本末倒置陷入死胡同。此外由于缺乏领域知识，对用户消费数据项的逻辑关系没有理清楚，导致最终排名不高。

此外对本次比赛还有一些小建议：

1. 比赛数据项解释并不清楚，特别是消费记录中的逻辑关系难以理解。
2. 比赛缺少分享机制，建议增加比赛论坛等，增加比赛过程经验分享。

最后感谢 rong360 举办的这次风控比赛，感谢比赛组织工作人员的辛勤工作！